OFFICE OF CIVILIAN RADIOACTIVE WASTE MANAGEMENT
ANALYSIS/MODEL COVER SHEET
Complete Only Applicable Items

1. QA: QA
Page: 1 of: 25

2. ☒ Analysis ☐ Engineering ☒ Performance Assessment ☐ Scientific	3. ☐ Model ☐ Conceptual Model Documentation ☐ Model Documentation ☐ Model Validation Documentation

4. Title:
Abstraction of Flow Fields for RIP (ID:U0125)

5. Document Identifier (including Rev. No. and Change No., if applicable):
ANL-NBS-HS-000023 REV 00

6. Total Attachments: 3	7. Attachment Numbers - No. of Pages in Each: I-17; II-4; III-7

	Printed Name	Signature	Date
8. Originator	Clifford K. Ho	Clifford K. Ho	1/19/2000
9. Checker	Yanyong Xiang	[illegible]	1/19/2000
10. Lead/Supervisor	Clifford K. Ho	Clifford K. Ho	1/19/2000
11. Responsible Manager	Clifford K. Ho	Clifford K. Ho	1/19/2000

12. Remarks:

OFFICE OF CIVILIAN RADIOACTIVE WASTE MANAGEMENT
ANALYSIS/MODEL REVISION RECORD
Complete Only Applicable Items

1. Page: 2 of: 25

2. Analysis or Model Title: Abstraction of Flow Fields for RIP (ID:U0125)	
3. Document Identifier (including Rev. No. and Change No., if applicable): ANL-NBS-HS-000023 REV 00	
4. Revision/Change No.	5. Description of Revision/Change
REV 00	Initial Issue

CONTENTS

Page

1. PURPOSE 5

2. QUALITY ASSURANCE 5

3. COMPUTER SOFTWARE AND MODEL USAGE 5

4. INPUTS 6
   4.1 DATA AND PARAMETERS 6
   4.2 CRITERIA 7
   4.3 CODES AND STANDARDS 8

5. ASSUMPTIONS 8

6. ANALYSIS/MODEL 9
   6.1 T2FEHM2 9
      6.1.1 Required Input Files for T2FEHM2 10
      6.1.2 Output Files from T2FEHM2 11
   6.2 WTRISE 20

7. CONCLUSIONS 22

8. REFERENCES 22
   8.1 DOCUMENTS CITED 22
   8.2 SOFTWARE 23
   8.3 SOURCE DATA 24

9. ATTACHMENTS 25
   I Listing of Software Routine T2FEHM2 v. 3.0 ..... I-1
   II Listing of Software Routine WTRISE v. 1.0 ..... II-1
   III Directory of files submitted to Technical Data Management System (DTN: SN9910T0581699.002) III-1

FIGURES

Page

Figure 1. 9-Node Example of the ncon Array Used in FEHM. 18

TABLES

Page

Table 1. Software Routines Used in this Analysis.. 6

Table 2. Input Data 6

1. PURPOSE

The purpose of this analysis is to post-process unsaturated-zone (UZ) site-scale flow fields that were simulated using TOUGH2 (Pruess 1991) by Lawrence Berkeley National Laboratory (LBNL). The resulting files will be used by the code FEHM (Zyvoloski et al. 1997) to perform particle tracking simulations for the total system performance assessment (TSPA) calculations. The TSPA code that integrates the FEHM particle tracking simulations with other aspects of performance assessment is called RIP (this name has recently been changed to GOLDSIM). The scope of this work is limited to the post-processing of eighteen "base-case" flow fields that are to be used by FEHM for particle tracking simulations in *Total System Performance Assessment—Site Recommendation Report* (TSPA-SR).

Constraints and limitations of this work include the preliminary status of the input data used in the analysis (see Section 4). Once these source data are qualified, the results of this analysis can be considered qualified. Until then, the information developed from this analysis must be considered unqualified. Detailed planning of this analysis can be found in the Development Plan, "Abstraction of Flow Fields for RIP (ID: U0125)" (CRWMS M&O 1999a).

2. QUALITY ASSURANCE

The Quality Assurance (QA) program applies to the development of this analysis and model report (AMR). The Performance Assessment Operations (PAO) responsible manager has evaluated this activity in accordance with QAP-2-0, *Conduct of Activities*. The QAP-2-0 activity evaluation (CRWMS M&O 1999b) determined that the development of this AMR is subject to the *Quality Assurance Requirements and Description* (DOE 1998) requirements.

3. COMPUTER SOFTWARE AND MODEL USAGE

Two software routines are used in this analysis and are summarized in Table 1. As part of this analysis, these routines are checked to ensure that they are providing correct results for the input files that are used (see Section 6). The use and documentation of these software routines comply with Section 5.1.1 of AP-SI.1Q (*Software Management*).

Table 1. Software Routines Used in this Analysis

Software Routine	Computer Platform	Comments
T2FEHM2 v. 3.0	Sun OS 5.7	This software routine post-processes the TOUGH2 flow fields to FEHM-readable files. It was compiled using FORTRAN 77 on the Sun OS 5.7 server (worf) at Sandia National Laboratories. The results of the output for this routine are verified by visual inspection within this analysis (Section 6). The listing of this routine is in Attachment I. All associated files with this software routine have been submitted to the Technical Data Management System (TDMS) (DTN: SN9910T0581699.002).
WTRISE v. 1.0	Sun OS 5.7	This software routine post-processes a FEHM '.ini' file to incorporate a water table rise. It was compiled using FORTRAN 90 on the Sun OS 5.7 server (worf) at Sandia National Laboratories. The results of the output for this routine are verified by visual inspection within this analysis (Section 6). The listing of this routine is in Attachment II. All associated files with this software routine have been submitted to the Technical Data Management System (TDMS) (DTN: SN9910T0581699.002).

4. INPUTS

The primary inputs to this analysis are the files associated with the LBNL flow fields that are to be used in TSPA-SR. These files were either transmitted from LBNL to PAO via AP-3.14Q (*Transmittal of Input*) or obtained from the TDMS as described below. The data are currently unqualified.

4.1 DATA AND PARAMETERS

Table 2 summarizes the input data used in this analysis. The data that was transmitted via Input Transmittal (CRWMS M&O 1999c,d) contained Data Tracking Numbers (DTNs) as shown in Table 2.

Table 2. Input Data

Description	Input Transmittal Number (ITN)	Data Tracking Number (DTN)	Comments
Flow Field Simulations and Infiltration Scenarios	SNL-LBL-99249.Ta	LB990801233129.001	Present day low infiltration; perched water model #1
		LB990801233129.002	Present day low infiltration; perched water model #2
		LB990801233129.003	Present day mean infiltration; perched water model #1
		LB990801233129.004	Present day mean infiltration; perched water model #2

Table 2. Input Data (Continued)

Description	Input Transmittal Number (ITN)	Data Tracking Number (DTN)	Comments
Flow Field Simulations and Infiltration Scenarios (Continued_	SNL-LBL-99249.Ta (Continued)	LB990801233129.005	Present day upper infiltration; perched water model #1
		LB990801233129.006	Present day upper infiltration; perched water model #2
		LB990801233129.007	Glacial transition low infiltration; perched water model #1
		LB990801233129.008	Glacial transition low infiltration; perched water model #2
		LB990801233129.009	Glacial transition mean infiltration; perched water model #1
		LB990801233129.010	Glacial transition mean infiltration; perched water model #2
		LB990801233129.011	Glacial transition upper infiltration; perched water model #1
		LB990801233129.012	Glacial transition upper infiltration; perched water model #2
		LB990801233129.013	Monsoon low infiltration; perched water model #1
		LB990801233129.014	Monsoon low infiltration; perched water model #2
		LB990801233129.015	Monsoon mean infiltration; perched water model #1
		LB990801233129.016	Monsoon mean infiltration; perched water model #2
		LB990801233129.017	Monsoon upper infiltration; perched water model #1
		LB990801233129.018	Monsoon upper infiltration; perched water model #2
Mesh	SNL-LBL-99249.Tb	LB990701233129.001	Mesh files used with flow fields. Files used in post-processing were mpa_pch1.v1 and mpa_pch2.v1.
Rock Grain Density	N/A	LB997141233129.001	The rock grain densities in the Excel spreadsheet in this DTN are used for the '.rock' file.

4.2 CRITERIA

No additional criteria govern the post-processing analysis beyond the scope and objectives presented in the Development Plan for this analysis (CRWMS M&O 1999a). Standard requirements are specified in AP-3.10Q (*Analyses and Models*). In addition, the Unsaturated Zone Flow and Transport Process Model Report and the TSPA-SR depend on the results of this analysis. These two reports have specific criteria as follows:

The U.S. Nuclear Regulatory Commission's (NRC's) Total System Performance Assessment and Integration (TSPA&I) Issue Resolution Status Report (IRSR) (NRC 1998) establishes generic technical acceptance criteria considered by the NRC staff to be essential to a defensible, transparent, and comprehensive assessment methodology for the repository system. These regulatory acceptance criteria address five fundamental elements of the U.S. Department of Energy (DOE) TSPA model for the Yucca Mountain site, namely:

1. Data and model justification (focusing on sufficiency of data to support the conceptual basis of the process model and abstractions);

2. Data uncertainty and verification (focusing on technical basis for bounding assumptions and statistical representations of uncertainties and parameter variabilities);

3. Model uncertainty (focusing on alternative conceptual models consistent with available site data);

4. Model verification (focusing on testing of model abstractions using detailed process-level models and empirical observations); and

5. Integration (focusing on appropriate and consistent coupling of model abstractions).

Relevant to the topic of this analysis, element (4) is addressed herein through the traceable documentation of the use of the UZ flow fields in TSPA calculations. Element (5) of the NRC acceptance criteria, which strictly applies to the completed synthesis of process-level models and abstractions, will be addressed separately in the TSPA-SR.

4.3 CODES AND STANDARDS

No codes or standards are applicable to the specific analysis in this report. However, the higher-level documents (e.g., TSPA-SR) that receive products from this report are intended to comply with the five NRC TSPA&I acceptance criteria, as well as the DOE interim guidance (Dyer 1999) which requires the use of specified Subparts/Sections of the proposed NRC high-level waste rule, 10 CFR Part 63 (64 FR 8640). Subparts of this proposed rule that are particularly applicable to data include Subpart B, Section 15 (Site Characterization) and Subpart E, Section 114 (Performance Assessment).

5. ASSUMPTIONS

- The rock grain densities for perched water elements in ‘pch1.rock’ and ‘pch2.rock’ are conservatively assumed to be equal to the lowest rock grain density in the parameter set (2240 kg/m^3). This assumption is necessary because rock grain densities are not listed for perched water elements in the source data. This assumption is conservative because it provides a lower bound to the retardation of radionuclides caused by sorption, which varies directly with rock grain density. (Section 6.1.2)

- Water-table rise does not significantly affect liquid mass flow rates and transport above the water table. This assumption allows the water table to be elevated in a post-processing routine without having to re-run the flow simulations (Section 6.2). The general flow pattern above the water table is primarily vertical (except in the perched water region, which will always provide lateral diversion due to its extremely low permeability). Therefore, the flow and transport is not expected to change significantly when the water table is simulated at a higher elevation.

- Mass sinks can be used below the prescribed water table in FEHM to signify locations where particles (radionuclides) leave the UZ and enter the SZ (Section 6.2). The mass sinks simply remove the particles from the "active" UZ domain and allow times to be recorded at which these particles leave the system. By placing these sinks at all nodes below the water table, the breakthrough times and mass flux of particles reaching the water table can be determined for input to saturated-zone studies.

6. ANALYSIS/MODEL

6.1 T2FEHM2

Simulations of unsaturated groundwater flow and radionuclide transport at Yucca Mountain, Nevada, are being performed for the TSPA-SR. Two numerical simulators are being used to perform the flow and transport simulations: (1) TOUGH2 (Pruess 1991) and (2) FEHM (Zyvoloski et al. 1997). Using various modules, both codes are capable of simulating hydrologic flow and radionuclide transport. However, a project decision was made to use TOUGH2 to simulate the hydrologic flow fields and to use FEHM to simulate the radionuclide transport. The TOUGH2 flow fields have the advantage of being calibrated to site data, and the FEHM particle tracking code has the advantage of being integrated with the TSPA code GOLDSIM (formerly RIP), in addition to being numerically efficient.

The method implemented by FEHM for UZ transport is called the Residence Time Transfer Function (RTTF) particle-tracking technique. It employs an efficient particle-tracking algorithm that eliminates the need to resolve the velocity vectors by interpolation at all particle positions. Instead, the mean residence time and the probabilities of travel to adjacent cells are computed for each cell. The information needed to compute particle residence times and pathways is readily available in finite-difference or finite-element solutions of the flow problem. The fluid mass and inter-cell mass flows rates are provided in the fluid flow solution, so the method can be implemented without regard to the nature of the numerical grid (structured versus unstructured grids, element shapes, etc.).

To facilitate the flow and transport simulation methodology, a post-processor was written to reformat TOUGH2 files that contain information pertaining to unsaturated flow to FEHM-readable files that can be used for radionuclide particle tracking. This method maintains consistency with the three-dimensional UZ site-scale flow fields that have been calibrated using TOUGH2.

FEHM uses a cell-based particle tracking model that preserves the overall residence time through any portion of the model and probabilistically reproduces the migration of a solute through the domain. The requirement for the method is that the flow calculation be based on a control volume in which fluid flow rates into and out of each cell are computed. Since TOUGH2 is an integrated finite difference code, and FEHM employs a control volume finite element technique, the two codes are compatible for implementing the particle tracking technique. The required inputs for FEHM to use an externally-developed flow field are: (1) grid connectivity information and cell volumes; (2) properties and state variables (rock grain density, fluid saturation, and rock porosity at each grid point); (3) inter-nodal fluid mass flow rate for every connection in the numerical grid; and (4) fluid source and sink flow rates for each grid block. The post-processor, T2FEHM2, was written to generate these required data from existing TOUGH2 files. The remainder of this section describes the required inputs to T2FEHM2 and the corresponding output files.

6.1.1 Required Input Files for T2FEHM2

When executed, T2FEHM2 will prompt the user for the names of three required files: (1) TOUGH2 input file; (2) TOUGH2 output file; and (3) TOUGH2 mesh file. T2FEHM2 will also prompt the user for the name of a fourth file containing the names of repository elements, but this file is optional. A T2FEHM2 input file that contains this information is included in each of the subdirectories submitted to the TDMS (DTN: SN9910T0581699.002).

TOUGH2 Input File

The TOUGH2 input file must contain the ROCKS and GENER cards. ROCKS contains material property information for fracture and matrix materials corresponding to a dual-permeability model. Fracture and matrix materials must have an 'F' or 'M', respectively, in the third or fourth character of the material name. Each material must have four lines associated with its entry. The GENER card should contain information on the infiltration source terms for prescribed elements. The generation rate is specified in units of kg/s.

TOUGH2 Output File

The TOUGH2 output file contains all simulated state variables (pressure, saturation) for each element and flux variables (mass flow rate) for each connection pair at user-specified print-out times. T2FEHM2 reads in these state and flux variables and puts them in a format that is compatible with FEHM.

TOUGH2 Mesh File

The TOUGH2 mesh file contains the ELEME and CONNE cards. ELEME contains the element names, material names, volumes, and coordinates of each element in the TOUGH2 model. The fracture and matrix elements should be listed alternately with a fracture element listed first. Also, all boundary elements must be listed at the end of the ELEME card. The material names associated with each element should be five-character names (not integers) that correspond identically to the name of one of the materials in the ROCKS card. The CONNE card contains all connection pairs and associated connection information for each element in the TOUGH2 model. T2FEHM2 stores these connection pairs to create connectivity arrays (ncon, istrw, nelmdg) for FEHM.

File Containing Repository Elements

A file containing the names of repository elements is optional. If present, T2FEHM2 will read the number of repository elements in the first line of the file. All repository element names will then be read from the file. These elements will be used to create special fracture and matrix zones in a FEHM file that will be used to define the location of radionuclide release for particle tracking.

6.1.2 Output Files from T2FEHM2

After reading the required information from the input files, T2FEHM2 prints out nine (9) files that are used by FEHM. The user specifies a reference file name, and the code creates nine output files by appending the following nine suffixes to the reference file name:

.dat
.dpdp
.files
.grid
.ini
.rock
.stor
.zone
.zone2

A tenth file, 'file_name.check,' is also printed but it is not used by FEHM. This file contains the node numbers and number of connections for each node. More detailed information on the contents of the FEHM macros can be found in Zyvoloski et al. (1997). The user should consult this information because a number of these macros have been created with T2FEHM2 using "dummy" variables that are either not needed by the particle tracking simulation (e.g., permeability, area coefficients, element specifications for nodes, etc.) or that can be modified by the user to suit the specific needs of the particle tracking simulation (e.g., date, time steps, print-out options, etc.). Most of these prescribed variables appear in the '*.dat' file, so the user should become familiar with the macros listed in that file before using the default values prescribed in T2FEHM2.

T2FEHM2 is run for each of the 18 flow fields shown in Table 2. The data submittal resulting from this analysis (DTN: SN9910T0581699.002) therefore contains T2FEHM2 output files in 18 subdirectories (the full directory listing is shown in Attachment III):

fm_pchl2: modern climate, lower infiltration, model #2
fm_pchm1: modern climate, mean infiltration, model #1
fm_pchm2: modern climate, mean infiltration, model #2
fm_pchu1: modern climate, upper infiltration, model #1
fm_pchu2: modern climate, upper infiltration, model #2

fm_monl1: monsoon climate, lower infiltration, model #1
fm_monl2: monsoon climate, lower infiltration, model #2
fm_monm1: monsoon climate, mean infiltration, model #1
fm_monm2: monsoon climate, mean infiltration, model #2
fm_monu1: monsoon climate, upper infiltration, model #1

fm_monu2: monsoon climate, upper infiltration, model #2

fm_glal1: glacial transition climate, lower infiltration, model #1
fm_glal2: glacial transition climate, lower infiltration, model #2
fm_glam1: glacial transition climate, mean infiltration, model #1
fm_glam2: glacial transition climate, mean infiltration, model #2
fm_glau1: glacial transition climate, upper infiltration, model #1
fm_glau2: glacial transition climate, upper infiltration, model #2

The prefix "fm" is placed in front of all T2FEHM2 files for identification purposes. The remainder of this section details the specific output files. To verify that T2FEHM2 is producing correct results, portions of the actual output files from 'fm_pchl1' (see above) are included. The values are compared to those in the original TOUGH2 files by visual inspection to ensure correct results.

Output File '*.dat'

This file contains the required macros used by FEHM: 'dpdp,' 'perm,' 'rlp,' 'rock,' 'flow,' 'time,' 'ctrl,' 'iter,' 'sol,' 'rflo,' 'air,' 'node,' 'zone,' 'ptrk.' If the macros are not explicitly defined in this file, the names of macro files containing the actual information are listed here. Macros 'perm' and 'rlp' are not required by the particle tracking solution, so dummy values are inserted here. In addition, many of the values in the '*.dat' file are prescribed within T2FEHM2 as default values, so the user should refer to Zyvoloski et al. (1997) to modify the values in the different macros to suit their needs. A sample file from 'fm_pchl1.dat' is provided below :

```
Low inf #1 perched water conceptual model  5/29/99 ysw
# Particle tracking for TOUGH2 flow field
dpdp
file
fm_pchl1.dpdp
perm
1  0  0   0.100E-14 0.100E-14 0.100E-14

rlp
1  0.  0.  1.  1.  0.  1.

1  0  0  1

rock
file
fm_pchl1.rock
flow

time
  0.36525E+09  0.36525E+09      10      10    1997      10

ctrl
     -10  0.10E-03      40
       1       0       0       1
0
      1.00      3.00      1.00
       5  0.20E+01  0.10E-09  0.10E+11
       0       1
iter
  0.10E-04  0.10E-04  0.10E-04 -0.10E-03  0.12E+01
       0       0       0       0  0.14E+05
sol
```

```
       1      -1
rflo
air
-1
20.0  0.1
node
1
1
zone
file
fm_pchl1.zone2
ptrk
file
fm_pchl1.ptrk
stop
```

Output File '*.dpdp'

This file contains a list of the zones corresponding to the fracture materials and lists the fracture porosities. It also contains dummy information regarding the length scale for matrix nodes that is not required for the TOUGH2-FEHM coupling. Here are the first few lines from 'fm_pchl1.dpdp' that can be compared to the ROCKS card used by TOUGH2:

```
dpdp
1
     -49       0       0     0.2800E-01
     -50       0       0     0.2000E-01
     -51       0       0     0.1500E-01
     -52       0       0     0.1100E-01
     -53       0       0     0.1200E-01
     -54       0       0     0.2500E-02
     -55       0       0     0.1200E-01
     -56       0       0     0.6200E-02
     -57       0       0     0.3600E-02
     -58       0       0     0.5500E-02
     -59       0       0     0.9500E-02
     -60       0       0     0.6600E-02
```

Output File '*.files'

This control file contains a list of files that FEHM reads for necessary information. Below is the 'fm_pchl1.files' file:

```
fm_pchl1.dat
fm_pchl1.grid
fm_pchl1.zone
fm_pchl1.out
fm_pchl1.ini
fm_pchl1.fin
fm_pchl1.his
fm_pchl1.trc
fm_pchl1.con

fm_pchl1.stor
fm_pchl1.chk
```

```
all
0
```

Output File '*.grid'

This file contains the coor and elem macros. The first line of the coor macro gives the total number of fracture elements, followed by a list of all the nodes in the fracture domain and their respective x, y, and z coordinates. The elem macro contains dummy information regarding the nodes associated with each element, but this is not required for the TOUGH2-FEHM coupling. Below are the first few lines of the 'fm_pchl1.grid' file that can be compared to the values in ELEME used by TOUGH2:

```
coor
   47664
       1    169398.60    236623.64      1626.10
       2    169398.60    236623.64      1606.47
       3    169398.60    236623.64      1569.84
       4    169398.60    236623.64      1547.73
       5    169398.60    236623.64      1535.45
       6    169398.60    236623.64      1519.98
       7    169398.60    236623.64      1496.52
       8    169398.60    236623.64      1464.89
```

Output File '*.ini'

This file contains re-start information for FEHM. The liquid saturations of all fracture and matrix nodes are listed following eight header lines. The gas-phase pressures (MPa) are then listed for the fracture and matrix nodes. The fourth header line ('air') tells FEHM that the pressures are for the gas phase. Then, mass flux values (kg/s) are listed for each connection of each node, starting with node 1 (the ordering is the same as the 'ncon' array in '.stor' without pointer information—see '*.stor' below). The mass flux values include sources (infiltration) denoted as negative values and sinks (connection to water table) denoted as positive values for each node. Flow into a node is negative, and flow out of a node is positive. The mass flux values for the fracture domain are listed first followed by the mass flux values in the matrix domain. The mass flux between fracture and matrix elements are listed last. Flow from the fracture to the matrix is denoted as positive. The first several lines of 'fm_pchl1.ini' is shown below:

```
Low inf #1 perched water conceptual model  5/29/99 ysw
This is a .ini file with saturations, pressures and mass flux values.
0.
air
ptrk
nstr
dpdp
ndua
  0.18430000E-01  0.12449000E-01  0.12477000E-01  0.33431000E-01
  0.13872000E-01  0.19117000E-01  0.12185000E-01  0.12171000E-01
  0.11877000E-01  0.10106000      0.17348000E-01  0.11237000E-01
```

Because there are 47664 active nodes in each of the fracture and matrix continua, we can find the appropriate line where the matrix saturations begin (47664 fracture nodes divided by 4 columns plus 8 header lines plus 1 line = 11925).

Line 11925 (first 4 matrix saturations):

```
0.33204000      0.78342000      0.71587000      0.55876000
```

The mass flux values can be found in the file by searching for "mass". It is found to be on line 47673, and several lines following that header are shown below (the values after the # symbol are not used by FEHM):

```
mass flux values
  830612     #ntotmfv=  782948, nnodes=   95328, number of f-m connections=    47664
-0.32450000E-03 0.32579000E-03-0.23469000E-05 0.20309000E-06-0.32579000E-03
         0.     0.33463000E-03-0.16918000E-04 0.51132000E-05 0.29560000E-05
-0.33463000E-03         0.     0.37063000E-03-0.42470000E-04 0.36439000E-05
 0.28288000E-05-0.37063000E-03         0.     0.14362000E-03-0.27325000E-05
```

Finally, the last few lines of the 'fm_pchl1.ini' are shown, which correspond to mass flow between the last few fracture/matrix connections:

```
0.29817000E-10 0.52044000E-12 0.18718000E-10 0.20154000E-10 0.31635000E-10
0.39433000E-10 0.17581000E-08 0.19384000E-08 0.21947000E-08 0.19911000E-08
0.17296000E-08 0.29018000E-09
```

Output File '*.rock'

This file lists the zones of all fracture and matrix materials. For each zone, the rock grain density (kg/m^3), specific heat (J/kg-K), matrix porosity, and intrinsic fracture porosity (1) are listed. The input files that were provided by LBNL contained incorrect rock grain densities because the values were not used in the UZ flow simulations (i.e., dummy values were used). However, rock grain densities are required for transport simulations to derive a retardation factor. Therefore, modified files of '*.rock' were created using the rock grain densities from the 1-D calibrated base-case parameter set (DTN: LB997141233129.001). For some materials in the LBNL UZ flow field input files, the materials were not listed in the calibrated parameter set (they were created to simulate perched water in the 3-D flow fields). For these materials, we conservatively used the lowest rock grain density in the parameter set (2240 kg/m^3). However, the values only range from a low of 2240 kg/m^3 to a high of 2580 kg/m^3, so not much impact is expected on using different values of the rock grain density. The files, 'pch1.rock' and 'pch2.rock' were created by hand and should supersede the '*.rock' files created by T2FEHM2. They are located in a separate subdirectory ('rock_files') of the data submittal for this analysis (see Attachment III).

Two '*.rock' files ('pch1.rock' and 'pch2.rock') are required to accommodate the two perched water models. Note that perched water model #1 has more materials that account for perched water elements. The first and last few lines from each are shown:

'pch1.rock'

```
rock
       -1    0     0     2550.  1.00E+03     2.53E-01
       -2    0     0     2510.  1.00E+03     8.20E-02
       -3    0     0     2470.  1.00E+03     2.03E-01
.
.
.
       -95   0     0     2240.  1.00E+03     1.00E+00
       -96   0     0     2240.  1.00E+03     1.00E+00
       -97   0     0     2.30E+03     1.00E+03     2.00E-01
       -98   0     0     2.30E+03     1.00E+03     2.80E-01

stop
```

'pch2.rock'

```
rock
       -1    0     0     2550. 1.00E+03    2.53E-01
       -2    0     0     2510. 1.00E+03    8.20E-02
       -3    0     0     2470. 1.00E+03    2.03E-01
.
.
.
       -86   0     0     2240. 1.00E+03    1.00E+00
       -87   0     0     2.30E+03    1.00E+03    2.00E-01
       -88   0     0     2.30E+03    1.00E+03    2.80E-01

stop
```

Output File '*.stor'

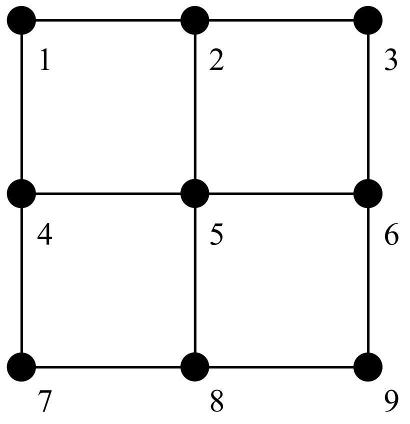
The file contains connectivity arrays and control volumes for the grid. Following two header lines, four integers are listed:

iwtotl:	Total number of connections in a continuum (either fracture or matrix) for which inter-node fluxes and areas are assigned. This includes connections for a node to itself for sources and sinks. Equal to ncont-(neq+1).
neq:	Number of nodes in either the fracture or matrix continuum.
ncont:	Number of values in the ncon array (see below)
sehtemp:	Flag that is equal to 1 for particle tracking

The following arrays are then read from .stor:

sx1(i), i=1,neq:	Primary (total) volume of each node in a continuum (includes fracture and matrix)

ncon(i), i=1,ncont:	Node connectivity array that contains the node numbers for each connection to a specified node in one continuum, starting with node 1. The node numbers in ncon associated with connections to a given node include the node of interest. All nodes connected to a given node are listed in ascending order. In the beginning of this array is pointer information with neq+1 entries. The entries identify the index of the array (i=1,ncont) that precedes the node denoted by the index of the pointer information. See Figure 1 for an example of a 9-node network.



index i	ncon(i)	
1	10	
2	13	
3	17	
4	20	
5	24	Pointer Information
6	29	
7	33	
8	36	
9	40	
10	43	
11	1	
12	2	Node 1
13	4	
14	1	
15	2	Node 2
16	3	
17	5	
18	2	
19	3	Node 3
20	6	
21	1	
22	4	Node 4
23	5	
24	7	
25	2	
26	4	
27	5	Node 5
28	6	
29	8	
30	3	
31	5	Node 6
32	6	
33	9	
34	4	
35	7	Node 7
36	8	
37	5	
38	7	Node 8
39	8	
40	9	
41	6	
42	8	Node 9
43	9	

Figure 1. 9-Node Example of the ncon Array Used in FEHM.

istrw(i), i=1,ncont: Not used in this application. The array is filled using following algorithm:

```
do i = 1, ncont
    if(i.le.iwtotl) then
        istrw(i) = i
    else
        istrw(i) = 0
    end if
end do
```

nelmdg(i), i=1,neq: Position (index) of node i in the ncon array:

```
do i = 1, neq
    do j = ncon(i) + 1, ncon(i+1)
        if (ncon(j).eq.i) nelmdg(i) = j
    end do
end do
```

iwtotl numbers: Three groups of iwtotl numbers signifying the x, y, and z components of the nodes are divided by distance terms for all internode connections. Only place-holders are required:

```
do i = 1,3
    write(15,'(5(1pe16.8))') (-1.0, j=1, iwtotl)
end do
```

The first few lines of the 'fm_pchl1.stor' files are shown:

```
Low inf #1 perched water conceptual model  5/29/99 ysw
This is a .stor file with dummy area coefficients
  391474     47664    439139         1
  2.33230000E+05  3.24388000E+06  3.24388000E+06  6.72396000E+05  1.50202000E+06
  1.23809500E+06  2.91903000E+06  2.68364000E+06  2.68364000E+06  1.07085600E+06
  1.77074000E+05  1.86573000E+06  2.70888000E+06  2.70888000E+06  2.91212000E+06
  4.35288000E+06  4.35288000E+06  1.38782000E+06  6.94010000E+05  1.28947000E+06
```

Output File '*.zone'

This file contains definitions of zones that correspond to ROCKS materials in TOUGH2. The materials are listed sequentially in the same order as they appear in the ROCKS card. A comment (#) is added to identify the name of the material as it appears in ROCKS. The number of nodes within each zone is listed after the header 'nnum'. Following that line, the nodes are listed in the order that they appear in the ELEME card in TOUGH2. Note that both the fracture and matrix materials are listed in this file. Additional comments are added after the 'stop' line of the file. The first few lines of 'fm_pchl1.zone' are shown:

```
zone
   1     #tcwM1
nnum
      278
     47665     47840     47934     47987     48081     48110     48186     48220
     48434     48493     48548     48608     48637     48687     48716     48833
     49175     49373     49440     49466     49527     49586     49890     50000
     50092     50145     50184     50308     50602     50974     51013     51215
```

Output File '*.zone2'

This file is identical to the .zone file except that it contains two additional zones that define the repository nodes for the fractures and matrix. The repository elements are listed in another file that is specified by the user during one of the prompts by T2FEHM2. This external file should contain the total number of repository elements in the file followed by a line-by-line listing of all the repository elements. This zone (.zone2) is read at the end of the .dat file to identify nodes where particles will be released in the ptrk macro (note that ptrk is not created by this post-processor). The nodes that are defined in .zone2 will retain the porosities and densities assigned to them previously in .rock and .dpdp. The file containing the repository nodes is called 'SR-repo-nodes' and is included in the data submittal to the TDMS (DTN: SN9910T0581699.002). The first few lines of 'SR-repo-nodes' are listed followed by the first few lines of the repository zone in 'fm_pchl1.zone2':

```
275
Fph 2
Foh 3
Fph 4
Foh 5


500   #fracture repository nodes
nnum
      275
     31351     31394     31439     31482     31525     31571     31615     31658
     31703     31748     31793     31839     31883     31927     31970     32014
```

6.2 WTRISE

The software routine, WTRISE, modifies the '.ini' file to simulate a water table rise to a user-prescribed elevation. This is required in TSPA calculations if the water table is assumed to rise above the modern-day elevation during future climate changes. For nodes below the prescribed water table, the liquid saturation is changed to one and a user-prescribed mass "sink" value is assigned to the node. This will remove particles from nodes that are "below" the prescribed water table during the FEHM particle tracking simulation. The post-processor can be summarized in the following steps (found in the header of the code listing in Attachment II):

```
c  (1) Read in user-defined files that contain the .grid, .stor, and
c      .ini files for FEHM, as well as the prescribed water-table rise
c      height (m) and "sink" value (kg/s)
c  (2) Read in nodes and coordinates from the .grid file
c  (3) Read in neq (number of fracture or matrix nodes), ncont, and
```

```
c      nelmdg values from .stor file.
c  (4) Read in .ini file and re-write new .ini file that has user-
c      prescribed sinks for nodes below prescribed water table height.
c      Also change saturations to one for nodes below water table.
```

Although the actual value of the future water table rise is uncertain, several studies have shown that the expected future water table rise at Yucca Mountain is at most 115 m (Quade et al. 1995, p. 213; Paces et al. 1993, p. 1573; and Marshall et al. 1993, p.1948). As a result, the value that was used in WTRISE for the runs in this analysis was a conservative rise of 120 m. Because the majority of the bottom boundary of the UZ model is near 730 m, the assumed elevation of the new water table was 850 m. WTRISE was run for each of the 18 flow fields, but the user will have to determine which climates are suitable for use with WTRISE (the present-day cases should not experience a rise in the current water table). Also, the water table elevation of 850 m can be easily changed by prescribing a different value and re-running WTRISE. The output files from the WTRISE runs have the suffix, ‘wt850.ini’. It should be noted that WTRISE is not required if a water table rise is not desired. Sinks are automatically implemented in T2FEHM2 for all elements with connections to the bottom boundary elements in the TOUGH2 model. A sample of the input file (wtrise.inp) for WTRISE is shown below:

```
fm_pchl1.grid
fm_pchl1.stor
fm_pchl1.ini
850
1.e10
fm_pchl1_wt850.ini
1

----------
      write(*,*)'What is the name of the .grid file?'
      read(*,*) grid
      write(*,*)'What is the name of the .stor file?'
      read(*,*) stor
      write(*,*)'What is the name of the .ini file?'
      read(*,*) ini
      write(*,*)'What is the desired water table elevation (m)?'
      read(*,*) wtr
      write(*,*)'What is the "sink" value for nodes below the'
      write(*,*)'water table (kg/s)?'
      read(*,*) sink
      write(*,*)'What is the desired name of the output file?'
      read(*,*) out
      write(*,*)'Print nodes below water table? (1=yes, 2=no)'
      read(*,*) ians
```

A few lines of the water-table-rise ‘.ini’ file are shown for ‘fm_pchl1_wt850.ini’ where “sinks” have been added to the nodes below the prescribed water table (the appropriate nodes can be checked by finding the line where the “mass” header line is located and finding the appropriate number of rows that must be added using the arrays in ‘.stor’):

Line 47713-47715:

```
0.12324000E-10-0.69223001E-05 0.10000000E+11 0.67923002E-05-0.65751998E-07
 0.15549000E-06-0.92942000E-11-0.67923002E-05 0.10000000E+11 0.67720998E-05
-0.76559999E-07 0.72921999E-07-0.75510001E-10-0.67720998E-05 0.10000000E+11
```

In addition, a file (wtnodes.dat) can be printed optionally from WTRISE to check that the nodes that have been modified are beneath the prescribed water table (this was done in the directory fm_pchl1).

7. CONCLUSIONS

This analysis has described the post-processing of 18 "base-case" unsaturated-zone site-scale flow fields developed by LBNL using the code TOUGH2. The software routine T2FEHM2 v. 3.0 was used to provide FEHM-readable files for use in TSPA particle tracking calculations. Another post-processor, WTRISE v. 1.0, was used to modify the post-processed flow fields in the '.ini' files to account for water-table rise. Nodes located beneath a prescribed water table elevation were modified to have a liquid saturation of one and a large (user-prescribed) mass "sink." Section 4 summarizes the inputs, and Section 6 provides details of the post-processing. All files associated with this analysis have been submitted to the TDMS (DTN: SN9910T0581699.002) and are considered "preliminary" pending qualification of upstream source data. Consequently, any use of data from this analysis is required to be controlled as "To Be Verified (TBV)" in accordance with appropriate procedures until this analysis becomes fully qualified.

8. REFERENCES

8.1 DOCUMENTS CITED

64 FR (Federal Register) 8640. *Disposal of High-Level Radioactive Wastes in a Proposed Geologic Repository at Yucca Mountain, Nevada*. Proposed rule 10 CFR 63. Readily Available.

AP-3.10Q, Rev. 1, ICN 1. *Analysis and Models*. Washington, D.C.: U.S. Department of Energy, Office of Civilian Radioactive Waste Management. ACC: MOL.19991019.0467.

AP-3.14Q, Rev. 0, ICN 0. *Transmittal of Input*. Washington, D.C.: U.S. Department of Energy, Office of Civilian Radioactive Waste Management. ACC: MOL.19990701.0621.

AP-SI.1Q, Rev. 2, ICN 1. *Software Management*. Washington, D.C.: U.S. Department of Energy, Office of Civilian Radioactive Waste Management. ACC: MOL.19991101.0212.

CRWMS M&O 1999a. *Abstraction of Flow Fields for RIP (ID: U0125)*. Development Plan TDP-NBS-HS-000042 REV 00. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19990811.0075.

CRWMS M&O 1999b. *Conduct of Performance Assessment*. Activity Evaluation. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19991028.0092.

DOE 1998. *Quality Assurance Requirements and Description.* DOE/RW-0333P, REV 8. Washington, D.C.: U.S. Department of Energy, Office of Civilian Radioactive Waste Management. ACC: MOL.19980601.0022.

Dyer, J.R. 1999. *Revised Interim Guidance Pending Issuance Of New U.S. Nuclear Regulatory Commission (NRC) Regulations (Revision 01, July 22, 1999), For Yucca Mountain, Nevada.* Letter from J. Russell Dyer (DOE) to D.R. Wilkins (YMP), September 3, 1999, OL&RC:SB-1714, with enclosure, "Interim Guidance Pending Issuance of New NRC Regulations for Yucca Mountain (Revision 01)." ACC: MOL.19990910.0079.

Marshall, B.D.; Peterman, Z.E.; and Stuckless, J.S. 1993. "Strontium Isotopic Evidence for a Higher Water Table at Yucca Mountain." *High Level Radioactive Waste Management, Proceedings of the Fourth Annual International Conference, Las Vegas, Nevada, April 26-30, 1993. 2*, 1948-1952. La Grange Park, Illinois: American Nuclear Society. TIC: 208542.

NRC 1998. *Issue Resolution Status Report Key Technical Issue: Total System Performance Assessment and Integration.* Rev. 1. Washington, D.C.: U.S. Nuclear Regulatory Commission. ACC: MOL.19990105.0083.

Paces, J.B; Taylor, E.M.; and Bush, C. 1993. "Late Quaternary History and Uranium Isotopic Compositions of Ground Water Discharge Deposits, Crater Flat, Nevada." *High Level Radioactive Waste Management, Proceedings of the Fourth Annual International Conference, Las Vegas, Nevada, April 26-30, 1993. 2*, 1573-1580. La Grange Park, Illinois: American Nuclear Society. TIC: 208542.

Pruess, K. 1991. *TOUGH2 - A General-Purpose Numerical Simulator for Multiphase Fluid and Heat Flow*. LBL-29400. Berkeley, California: Lawrence Berkeley National Laboratory. ACC: NNA 19940202.0088.

Quade, J; Mifflin, M.D.; Pratt, W.L.; and McCoy, W. 1995. "Fossil Spring Deposits in the Southern Great Basin and Their Implications for Changes in Water-Table Levels Near Yucca Mountain, Nevada, During Quaternary Time." *GSA Bulletin*, v. 107, no. 2, 213-230. TIC: 234256.

QAP-2-0, Rev. 5, ICN 0. *Conduct of Activities*. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19980826.0209.

Zyvoloski, G.A.; Robinson, B.A.; Dash, Z.V.; and Trease, L.L. 1997. *User's Manual for the FEHM Application - A Finite-Element Heat-and-Mass-Transfer Code.* LA-13306-M. Los Alamos, New Mexico: Los Alamos National Laboratory. TIC: 235999.

8.2 SOFTWARE

Only software routines are used, and they are documented as part of this analysis (see Section 3, Table 1 for details).

8.3 SOURCE DATA

CRWMS M&O 1999c. *Flow Field Simulation and Infiltration Scenarios*. Input Transmittal SNL-LBL-99249.Ta. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19990930.0112. (see Table 2 for list of DTNs associated with this Input Transmittal).

CRWMS M&O 1999d. *Mesh Used with Flow Field Simulation*. Input Transmittal SNL-LBL-99249.Tb. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19991015.0298. (see Table 2 for DTN associated with this Input Transmittal).

LB990701233129.001. 3-D UZ Model Grids for Calculation of Flow Fields for PA for AMR U0000, "Development of Numerical Grids for UZ Flow and Transport Modeling." Submittal date: 09/24/1999. (TBV-3167)

LB990801233129.001. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #1). Submittal date: 11/29/1999. (TBV-3660)

LB990801233129.002. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #2). Submittal date: 11/29/1999. (TBV-3661)

LB990801233129.003. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #3). Submittal date: 11/29/1999. (TBV-3662)

LB990801233129.004. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #4). Submittal date: 11/29/1999. (TBV-3663)

LB990801233129.005. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #5). Submittal date: 11/29/1999. (TBV-3664)

LB990801233129.006. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #6). Submittal date: 11/29/1999. (TBV-3665)

LB990801233129.007. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #7). Submittal date: 11/29/1999. (TBV-3666)

LB990801233129.008. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #8). Submittal date: 11/29/1999. (TBV-3667)

LB990801233129.009. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #9). Submittal date: 11/29/1999. (TBV-3668)

LB990801233129.010. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #10). Submittal date: 11/29/1999. (TBV-3669)

LB990801233129.011. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #11). Submittal date: 11/29/1999. (TBV-3670)

LB990801233129.012. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #12). Submittal date: 11/29/1999. (TBV-3671)

LB990801233129.013. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #13). Submittal date: 11/29/1999. (TBV-3672)

LB990801233129.014. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #14). Submittal date: 11/29/1999. (TBV-3673)

LB990801233129.015. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #15). Submittal date: 11/29/1999. (TBV-3674)

LB990801233129.016. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #16). Submittal date: 11/29/1999. (TBV-3675)

LB990801233129.017. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #17). Submittal date: 11/29/1999. (TBV-3676)

LB990801233129.018. TSPA Grid Flow Simulations for AMR U0050, "UZ Flow Models and Submodels." (Flow Field #18). Submittal date: 11/29/1999. (TBV-3677)

LB997141233129.001. Calibrated Basecase Infiltration 1-D Parameter Set for the UZ Flow and Transport Model, FY99. Submittal date: 07/21/1999. (TBV-3095)

9. ATTACHMENTS

Attachment	Title
I	Listing of Software Routine T2FEHM2 v. 3.0
II	Listing of Software Routine WTRISE v. 1.0
III	Directory of files submitted to Technical Data Management System (DTN: SN9910T0581699.002)

ATTACHMENT I

Listing of Software Routine T2FEHM2 v. 3.0

```
c     t2fehm2_v3.f
C*****************************************************************
c     This program creates column formatted files from TOUGH2.OUT
c     files of EOS3 simulations.
c     Files  MESH, TOUGH2.INP, and TOUGH2.OUT must be present.
c     The format of the output files are amenable for an FEHM
c     restart.
c                        C.K.Ho 5/27/97
c  This version now re-formats TOUGH2.OUT files in either EOS3 or
c  EOS9 format. Multidimensional files can be post-processed. This
c  version assumes that the elements listed in ELEME alternate
c  between fractures and matrix, starting with a fracture element.
c  This can be generalized in the loop (do 3000...) by knowing how
c  how the fracture and matrix elements were listed and by arranging
c  the arrays accordingly.  I started this by asking the user to
c  specify the ordering, but I didn't do much with it in this version.
c  So for now, the elements should be listed alternately starting with
c  a fracture element. Also, the matrix materials are assumed to be listed
c  first in the ROCKS card.
c
c     C.K.Ho
c     9/2/97-9/12/97,9/19/97
c  This version (op1postv3.f) is tailored specifically for LBL site-scale
c  runs. The previous version (option1postv2.f) is still good for SNL
c  TOUGH2 simulations of flow fields.  The major revisions include reading
c  information from external files (MESH, GENER). In MESH, the material
c  identifier is a 5-character name--not an integer, which was assumed in
c  the previous version.  The coordinates will have to be
c  read from MESH. Changes will have to be made for recognizing
c  fracture or matrix materials to accomodate all the materials (there
c  are greater than 100 materials) in the site-scale model. The dimensions
c  will have to be greatly increased to accommodate the 80,000 element
c  site-scale model.
c     C.K.Ho
c     10/23/97
c
c  This version (op1postv4.f) does not assume any ordering in the ROCKS
c  card.  There can be different numbers of matrix and fracture
c  materials written to the FEHM zone macro.  Also, this version can read
c  in a file containing repository element names to create a separate zone.
c  Another assumption is that the active elements are listed before any
c  boundary elements ('TP' or 'BT') in ELEME.
c     C.K.Ho
c     11/5/97
c
c  A few things have been cleaned up and it appears to work for the LBNL
c  3-D site scale model.  The current version is 't2fehm2.f'.
c     C.K.Ho
c     11/6/97
c
c  This version accommodates new output formatting used by LBNL.  The
c  index field in the output has been changed from i6 to i12.  Also,
c  the flux output has been shifted to the left a bit, and nlin3 is now equal
c  to 3 instead of 4 (this is the amount of header lines inserted in the flux
c  output periodically).
c  The liquid pressure now appears where the gas pressure used to appear in
c  the output file.  To calculate the gas pressure: Pg=Pl-Pc
c     C.K.Ho
c     3/9/99
c
c  This version allows the user to calculate the primary volumes as the
c  sum of the fracture and matrix volumes listed in the TOUGH2 ELEME card.
c  This eliminates the need to divide the fracture volume in the ELEME
c  card by the fracture porosity in the ROCKS card, which in the SR
c  base-case runs have in some instances been tranformed to matrix
c  porosities for perched water materials, yielding incorrect primary
c  volumes.
c     C.K.Ho
c     10/10/99
c*****************************************************************
c234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
      implicit double precision (a-h,o-z)
      DIMENSION X(99000),Y(99000),z(99000),SL(99000),vol(99000)
      dimension PG(99000)
      dimension gelem(99000),ifm(99000)
      dimension fluxl(990000),fmlfm(99000),ncord(99000)
      dimension icon2(990000),flol2(990000),istrw(990000)
      dimension drok(500),por(500),nelmdg(99000),ncon2(99000)
      double precision lblpor
      CHARACTER*22 BLOCK
      CHARACTER*5  ELEMN(99000),ELEM1(490000),ELEM2(490000),ELEMX
      character*5  genname,matname(500),matb,mat(99000)
      character*80 header
      character*40 filen,control,dat,grid,ini,stor,dpdp,rock,zone
      character*40 filein,fileout,meshfile,repfile,zone2,check
      character*1 char2
      character*5 repname(1003)
      common/int/ ncon(99000),icon(99000,35)
      common/flux/ flol(99000,35)
C
      write(*,*) 'This program will re-format TOUGH2 output files'
      write(*,*)'for FEHM restart files. The following files'
      write(*,*)'must be present: input, output, and MESH.'
      write(*,*)'The MESH file should contain 5-character material'
      write(*,*)'names.'
      write(*,*)
      write(*,*)'What is the name of the input file?'
      read(*,*) filein
      write(*,*)'What is the name of the output file?'
      read(*,*) fileout
      write(*,*)'What is the name of the MESH file?'
      read(*,*) meshfile
      write(*,4)
4     format('What type of run is this?'/,'1) SNL EOS3'/,'2) SNL EOS9?'/
     & ,'3) LBNL EOS9'/,'4) LBNL EOS9 SR/LA')
      read(*,*) neos
      write(*,*)'What reference name would you like to use for the'
      write(*,*)'FEHM restart files? (no spaces in the name)'
      read(*,*) filen
      write(*,*)'In ELEME, how are the elements listed?'
      write(*,*)'(1) Alternatively with matrix first'
      write(*,*)'(2) Alternatively with fracture first'
      write(*,*)'(3) All matrix, then all fractures'
      write(*,*)'(4) All fractures, then all matrix'
      read(*,*) norder
      write(*,*)'For fracture-matrix connections, which element is'
      write(*,*)'listed first:  (1) Fracture or (2) Matrix?'
      read(*,*) nfmc
      write(*,*)'What is the print-out time (sec) of interest?'
      read(*,*) tsec
      write(*,*)'Would you like to calculate primary volumes as the'
      write(*,*)'sum of the fracture & matrix volumes in ELEME?'
      write(*,*)'(1=yes, 0=no)'
      read(*,*) nprimary
      volscale=1.
      if(nprimary.eq.1) go to 6
      write(*,*)'The fracture volumes will be used as the primary'
      write(*,*)'control volume for each element.  Have they been'
      write(*,*)'modified in TOUGH2.INP? (1=yes,  0=no)'
      read(*,*) nvol
      if(nvol.eq.1) then
        write(*,*)'What is the scaling factor to retrieve correct',
     &  ' primary volumes from fracture volumes?'
        read(*,*) volscale
      end if
6     write(*,7)
7     format('What is the geometry?'/'0) 3-D'/'1) X-Y Plane'/
     &       '2) X-Z Plane'/'3) Y-Z Plane')
      read(*,*) icnl
      write(*,*)'Is there a file with repository element names?'
      write(*,*)'1 = yes,  0 = no'
      read(*,*) nrepans
      if(nrepans.eq.1) then
        write(*,9)
9     format('What is the name of the file with repository elements?')
        read(*,*) repfile
        write(*,*)'Would you like to modify the 2nd character of the'
        write(*,*)'element name?  1=yes,  0=no'
        read(*,*) n2nd
        if(n2nd.eq.1) then
          write(*,*)'What character would you like to use?'
          read(*,'(a1)') char2
        end if
        open(19,file=repfile,status='old')
      end if

      if(norder.eq.1.or.norder.eq.2) then
        nalt=2
      else
        nalt=1
      end if

c...Define FEHM restart files based on reference name
      kend=index(filen,' ')
      control=filen(1:kend-1)//'.files'
      dat=filen(1:kend-1)//'.dat'
      grid=filen(1:kend-1)//'.grid'
      ini=filen(1:kend-1)//'.ini'
      stor=filen(1:kend-1)//'.stor'
      dpdp=filen(1:kend-1)//'.dpdp'
      rock=filen(1:kend-1)//'.rock'
      zone=filen(1:kend-1)//'.zone'
      zone2=filen(1:kend-1)//'.zone2'
      check=filen(1:kend-1)//'.check'

      if(neos.eq.1) then
        nlin1=5
        nlin2=3
        nlin3=3
      elseif(neos.eq.2) then
        nlin1=6
        nlin2=4
        nlin3=4
      elseif(neos.eq.3) then
        nlin1=6
        nlin2=3
        nlin3=4
      elseif(neos.eq.4) then
        nlin1=6
        nlin2=3
        nlin3=3
      end if

      write(*,*) 'Thank You!  Please wait while I work...'
      open(1,file=meshfile,status='old')
      open(2,file=fileout,status='old')
      open(3,file=filein,status='old')
      open(11,file=control,status='unknown')
      open(12,file=dat,status='unknown')
      open(13,file=grid,status='unknown')
      open(14,file=ini,status='unknown')
      open(15,file=stor,status='unknown')
      open(16,file=dpdp,status='unknown')
      open(17,file=rock,status='unknown')
      open(18,file=zone,status='unknown')
      open(22,file=check,status='unknown')
      open(23,file=zone2,status='unknown')

c....Data
      spht=1.e3
      per1=1.e-15
      per2=1.e-15
      per3=1.e-15
      day=365.25e6
      tims=365.25e6
      nstep=10
      iprtout=10
      iyear=1997
      month=10
      maxit=-10
      epm=1.e-4
      north=40
      ja=1
      jb=0
      jc=0
      igaus=1
      as=1.
      grav=3.
      upwgt=1.
      iamm=5
      aiaa=2.
      daymin=1.e-10
      daymax=1.e10
      lda=1
      g1=1.e-5
      g2=1.e-5
      g3=1.e-5
      tmch=-1.e-4
      overf=1.2
      irdof=0
      islord=0
      iback=0
      icoupl=0
      rnmax=14400.
      ntt=1
      intg=-1
      zero=1.d-10
      ra=287.
      rv=461.52
C
c...Read header from TOUGH2.INP
      read(3,'(a80)') header

c_______________________________________________________________
c...Write information to .dat file
      write(12,510) header
510   format(a80/'# Particle tracking for TOUGH2 flow field')

c...Write dpdp macro
      write(12,516) dpdp
516   format('dpdp'/'file'/a)

c...Write perm macro
      write(12,518) per1,per2,per3
518   format('perm'/'1  0  0  ',3e10.3/)

c...Write rlp macro
      write(12,520)
520   format('rlp'/'1  0.  0.  1.  1.  0.  1.'//'1  0  0  1'/)

c...Write rock macro
      write(12,522) rock
522   format('rock'/'file'/a)

c...Write flow macro
      write(12,524)
524   format('flow'/)

c...Write time macro
      write(12,526) day,tims,nstep,iprtout,iyear,month
526   format('time'/2e13.5,4i8/)

c...Write ctrl macro
      write(12,528) maxit,epm,north,ja,jb,jc,igaus,as,grav,upwgt,
     & iamm,aiaa,daymin,daymax,icnl,lda
528   format('ctrl'/i8,e10.2,i8/4i8/'0'/3f10.2/i8,3e10.2/2i8)

c...Write iter macro
      write(12,530) g1,g2,g3,tmch,overf,irdof,islord,iback,icoupl,
     & rnmax
530   format('iter'/5e10.2/4i8,e10.2)

c...Write sol macro
      write(12,532) ntt,intg
532   format('sol'/2i8)

c...Write rflo macro
      write(12,534)
534   format('rflo'/'air'/'-1'/'20.0  0.1')

c...Write node macro
      write(12,536)
536   format('node'/'1'/'1')

c...Write zone macro that corresponds to the repository nodes
```

```
      write(12,515) zone2
515   format('zone'/'file'/a)

c...Write ptrk macro
      write(12,538) filen(1:kend-1)
538   format('ptrk'/'file'/a,'.ptrk')

c...Write stop
      write(12,540)
540   format('stop')
c_______________________________________________________________

c...Write information to control file
      write(11,501) dat,grid,zone,filen(1:kend-1),ini,filen(1:kend-1)
     &,filen(1:kend-1),filen(1:kend-1),filen(1:kend-1),stor,
     &filen(1:kend-1)
501   format(a/a/a/a,'.out'/a/a,'.fin'/a,'.his'/a,'.trc'/a,'.con'//
     & a/a,'.chk'/'all'/'0')

c...Read in repository element names
      if(nrepans.eq.1) then
        read(19,*) nrepelem
        numrep=nrepelem
        do i=1,nrepelem
          read(19,'(a5)') repname(i)
          repname(i)(1:1)='F'
          if(n2nd.eq.1) repname(i)(2:2)=char2
        end do
      end if

c...Read in grid information from MESH
      nbelm=0
      nbmat=0
      matb='     '
      N=1
      read(1,1000) block
1000  format(a22)
99    read(1,65) elemn(n),mat(n),vol(n),x(n),y(n),z(n)
65    format(a5,10x,a5,e10.4,20x,3e10.4)
      if(elemn(n).eq.'     ') go to 98
      if(elemn(n)(4:4).eq.'0') elemn(n)(4:4)=' '
c...Count number of boundary elements, nbelm, and number of boundary
c...materials, nbmat.
      if(elemn(n)(1:2).eq.'TP'.or.elemn(n)(1:2).eq.'BT') then
        nbelm=nbelm+1
        if(mat(n).ne.matb) then
          nbmat=nbmat+1
          matb=mat(n)
        end if
      end if
      N=N+1
      GO TO 99
98    CONTINUE
      NMAX = N - 1
c...NMAX is the total number of elements read from MESH
      write(*,107) nmax
107   format('Have read in ',i8,' elements from MESH...')
c...nnodes is the total number of active nodes
      nnodes=nmax-nbelm

c...Find maximum number of materials used in ROCKS (nmat)
c      nmat=0
c      do i=1,nmax
```

```
c         nmat=max(mat(i),nmat)
c       end do
c       write(*,222) nmat
c222    format('Maximum number of active materials = ',i8,'...')

c...nfmat is the number of fracture materials
c       nfmat=(nmat-nbmat)/2

c...Read in connection information from MESH
      N=1
      READ(1,1500) BLOCK
1500  FORMAT(A22,3X,25X,E10.4)
199   read(1,1502) elem1(n),elem2(n),ifm(n)
c...ifm(n) is a flag in the 75th column of the CONNE card that Yu-Shu has
c...specified as equal to '2' for fracture-matrix connections
1502  format(2a5,64x,i1)
      IF(elem1(n)(1:5).EQ.'     '.OR.elem1(n)(1:3).EQ.'+++') GO TO 198
      if(elem1(n)(4:4).eq.'0') elem1(n)(4:4)=' '
      if(elem2(n)(4:4).eq.'0') elem2(n)(4:4)=' '
      N=N+1
      GO TO 199
198   CONTINUE
      NCMAX = N - 1
c...NCMAX is the total number of connections read from MESH
      write(*,203) ncmax
203   format('Have read in ',i8,' connections from MESH...')

c...Read in ROCKS information from TOUGH2 input file
18    read(3,1000) block
      if(block(1:5).ne.'ROCKS') go to 18

      i=1
      nfmat=0
      nmmat=0
408   read(3,410) matname(i),drok(i),por(i)
410   format(a5,5x,2e10.4)
      if(matname(i).eq.'REFCO') go to 408
      if(matname(i).eq.'     ') then
c...ntotmat is the total number of materials in the ROCKS card
c...nmat is the number of materials associated with non-boundary
c...elements
        ntotmat=i-1
        nmat=ntotmat-nbmat
        go to 27
      end if
c...LBNL uses columns 71-80 in the second line of each material card to
c...identify the fracture porosity
      read(3,415) lblpor
415   format(70x,e10.4)
c...nfmat is the total number of fracture materials
      if(matname(i)(3:3).eq.'F'.or.matname(i)(4:4).eq.'F') then
        nfmat=nfmat+1
        if(neos.eq.3.or.neos.eq.4) por(i)=lblpor
c...The perched water fractures do not have porosities listed in ROCKS.
c...Yu-Shu said that they have the same porosity as the zeolitic fractures,
c...which is 1.1e-5 (phone message 10/31/97).
        if(por(i).eq.0.) por(i)=1.1d-5
      end if
c...nmmat is the total number of matrix materials
      if(matname(i)(3:3).eq.'M'.or.matname(i)(4:4).eq.'M') nmmat=nmmat+1
      read(3,*)
      read(3,*)
      i=i+1
      go to 408

27    continue

c...10/27/97  Ho

c...Write grid macro file
      write(13,202) nnodes/2
202   format('coor'/i8)
c...This assumes that all boundary elements ('TP' and 'BT') are listed
c...after the active elements in ELEME
      do i=1,nnodes/2
        write(13,204) i,x(i*nalt),y(i*nalt),z(i*nalt)
204     format(i8,3(3x,f10.2))
      end do
      write(13,206)
206   format(/'elem'/'2  1'/'1  2  1'//'stop')

c...Initialize generation array
      do i=1,nmax
        gelem(i)=0.
      end do

c...Read in generation information from TOUGH2.INP
      i=1
33    read(3,1000,end=299) block
      if(block(1:5).ne.'GENER') go to 33
74    read(3,75) genname,g
75    format(a5,35x,e10.4)
      if(genname.eq.'     ') go to 77
      if(genname(4:4).eq.'0') genname(4:4)=' '
      do ik=1,nmax
       if(genname.eq.elemn(ik)) then
c...Assign a generation term for each element (flow into an element
c...is defined as negative)
c...The method used here is different than in v3.  It eliminates a
c...separate do-loop and the need for arrays igen and g.
         gelem(ik)=-g
         i=i+1
         go to 74
       end if
      end do
      write(*,*)'Could not find element name for generation'
      write(*,79) i,genname
79    format('element ',i8,': ',a5)
      stop

299   write(*,*)'***Warning*** No generation card in TOUGH2.INP'

77    ngentot=i-1

c...Write zone macro
      ntotin=0
      write(18,'(a4)') 'zone'
      write(23,'(a4)') 'zone'
      do i=1,ntotmat
        write(18,512) i,matname(i)
        write(23,512) i,matname(i)
512     format(i4,5x,'#',a5)
        write(18,'(a4)') 'nnum'
        write(23,'(a4)') 'nnum'
        nin=1
        do j=1,nmax
```

```
c...Match nodes to respective materials. This assumes that the
c...fractures and matrix elements are listed alternately in ELEME
c...starting with the fractures first
c...If element is a boundary element, go to next element
          if(elemn(j)(1:2).eq.'TP'.or.elemn(j)(1:2).eq.'BT') goto 517
          if(mat(j).eq.matname(i)) then
            if(mat(j)(3:3).eq.'F'.or.mat(j)(4:4).eq.'F') then
               ncord(nin)=(j+1)/nalt
               nin=nin+1
               go to 517
            end if
            if(mat(j)(3:3).eq.'M'.or.mat(j)(4:4).eq.'M') then
              ncord(nin)=j/nalt+nnodes/2.
              nin=nin+1
            end if
          end if
517     end do
        nin=nin-1
        ntotin=ntotin+nin
        write(18,'(i10)') nin
        write(23,'(i10)') nin
        if(nin.gt.0) write(18,'(8i10)') (ncord(k),k=1,nin)
        if(nin.gt.0) write(23,'(8i10)') (ncord(k),k=1,nin)
      end do
      write(18,*)
      write(18,'(a4)') 'stop'

c...Now write zones for nodes corresponding to repository elements
      nrp=1
      do i=1,nmax
        do j=1,numrep
          if(elemn(i).eq.repname(j)) then
            ncord(nrp)=(i+1)/nalt
            nrp=nrp+1
            go to 527
          end if
        end do
527   end do

      nrp=nrp-1
      write(23,*) '500   #fracture repository nodes'
      write(23,'(a4)') 'nnum'
      write(23,'(i10)') nrp
      if(nrp.gt.0) write(23,'(8i10)') (ncord(k),k=1,nrp)
      write(23,*) '501   #matrix repository nodes'
      write(23,'(a4)') 'nnum'
      write(23,'(i10)') nrp
      do i=1,nrp
        ncord(i)=ncord(i)+nnodes/2.
      end do
      if(nrp.gt.0) write(23,'(8i10)') (ncord(k),k=1,nrp)
      write(23,*)
      write(23,'(a4)') 'stop'

c...Now write some additional information to the zone file
      write(18,*)
      write(23,*)
      write(18,514) ntotin,nbmat,nbelm
      write(23,514) ntotin,nbmat,nbelm
514   format(/'#Total number of nodes = ',i8/'#Total number of',
     & ' active boundary materials = ',i8/'#Total number of active',
     & ' boundary nodes = ',i8/)
```

```
c...Write dpdp macro file
      write(16,550)
550   format('dpdp'/'1')
c...Loop over the materials and print out fracture porosities
      do i=1,ntotmat
        if(matname(i)(3:3).eq.'F'.or.matname(i)(4:4).eq.'F') then
          write(16,552) -i,jb,jc,por(i)
552       format(3i8,5x,e10.4)
        end if
      end do
      write(16,554) ja,jb,jc
554   format(/,3i8,5x,'99.'//'stop')

c...Write rock macro file
      write(17,556)
556   format('rock')
      do i=1,ntotmat
        porock=por(i)
        if(matname(i)(3:3).eq.'F'.or.matname(i)(4:4).eq.'F')porock=1.
        write(17,558) -i,jb,jc,drok(i),spht,porock
558     format(3i8,5x,e10.4,5x,e10.4,5x,e10.4)
      end do
      write(17,559)
559   format(/'stop')

c...Search for "TOTAL TIME" in TOUGH2.OUT and then read in variables
89    READ(2,1000,END=90) BLOCK
      IF(BLOCK(1:12).NE.'  TOTAL TIME') GO TO 89
      READ(2,1001) TIME
      if(time.ne.tsec.and.tsec.gt.0) go to 89
1001  FORMAT(E13.5)
      do nl=1,nlin1
        READ(2,1000) BLOCK
      end do
C
c234567890123456789012345678901234567890123456789012345678901234567890l2

c...Read in state variables from TOUGH2.OUT
115   N1=1
      N2=MIN(NMAX,45)
      DO 2000 I=N1,N2
      if(neos.eq.1) then
c...    This is EOS3 format
        READ(2,1002) PG(I),SL(I)
1002    FORMAT(12x,e12.5,24x,7e12.5)
      elseif(neos.eq.2.or.neos.eq.3) then
c...    This is EOS9 format
        read(2,118) pg(i),sl(i)
118     format(12x,2e12.5)
      elseif (neos.eq.4) then
c...    This is EOS9 format with new index formatting of i12
        read(2,119) pl,sl(i),pc
        pg(i)=pl-pc
119     format(18x,3e12.5)
      end if
2000  CONTINUE
C
2100  CONTINUE
c...Check to see if we've read in all the element variables
      IF(N2.EQ.NMAX) GO TO 91
      N1=N2+1
      N2=MIN(NMAX,N1+56)
      do nl=1,nlin2
```

```
          READ(2,1000) BLOCK
        end do
        DO 2010 I=N1,N2
        if(neos.eq.1) then
c...      This is EOS3 format
          READ(2,1002) PG(I),SL(I)
        elseif(neos.eq.2.or.neos.eq.3) then
c...      This is EOS9 format
          read(2,118) pg(i),sl(i)
        elseif (neos.eq.4) then
c...      This is EOS9 format with new index formatting of i12
          read(2,119) pl,sl(i),pc
          pg(i)=pl-pc
        end if
2010    CONTINUE
        GO TO 2100
C
91      CONTINUE
C
c...Write saturations to .ini file (fractures saturations first followed
c...by matrix saturations)
        write(14,302) header
302     format(a80/'This is a .ini file with saturations, pressures',
       & ' and mass flux values.'/'0.'/'air'/'ptrk'/'nstr'/
       & 'dpdp'/'ndua')
        write(14,304) (sl(i),i=1,nnodes,2),(sl(i),i=2,nnodes,2)
304     format(4g16.8)

c...Write pressures to .ini file in MPa (fractures first, then matrix)
        write(14,304) (pg(i)*1.d-6,i=1,nnodes,2),
       & (pg(i)*1.d-6,i=2,nnodes,2)

        write(*,*)'Have read in state variables from output file...'
C
c...Read in flux variables from TOUGH2.OUT
289     READ(2,1500,END=190) BLOCK
        if(neos.lt.4) then
          IF(BLOCK(11:22).NE.'ELEM1  ELEM2') GO TO 289
        elseif (neos.eq.4) then
          IF(BLOCK(7:18).NE.'ELEM1  ELEM2') GO TO 289
        end if
        READ(2,1500) BLOCK
        READ(2,1500) BLOCK
C
c...Read in mass flow liquid for each connection pair
        N1=1
        N2=MIN(NCMAX,53)
        DO 1600 I=N1,N2
        if(neos.eq.1) then
          READ(2,1003) fluxl(I)
1003      FORMAT(80x,4e13.5)
        elseif (neos.eq.2.or.neos.eq.3) then
          read(2,121) fluxl(i)
121       format(29x,e13.5)
        elseif (neos.eq.4) then
          read(2,122) fluxl(i)
122       format(31x,e13.5)
        end if
1600    CONTINUE
C
2150    CONTINUE
        IF(N2.EQ.NCMAX) GO TO 191
        N1=N2+1
```

```
      N2=MIN(NCMAX,N1+56)
      do nl=1,nlin3
        READ(2,1500) BLOCK
      end do
      DO 2020 I=N1,N2
      if(neos.eq.1) then
        READ(2,1003) fluxl(I)
      elseif (neos.eq.2.or.neos.eq.3) then
        read(2,121) fluxl(i)
      elseif (neos.eq.4) then
        read(2,122) fluxl(i)
      end if
2020  CONTINUE
      GO TO 2150
C
191   CONTINUE

C
190   CONTINUE

c...Check
      write(*,*)'Have read in flux variables from output file...'

c...Check
c      do i=1,ncmax
c        write(15,444) i,elem1(i),elem2(i),fluxl(i)
c444     format(i8,2x,2(a5,2x),e10.4)
c      end do
c      stop
c...End check
C
c...Loop over all elements to determine connections and fluxes for each
c...element
      nmlfm=1
c...nmlfm is the total number of fracture-matrix connections
      DO 3000 I=1,NMAX

       if(mod(i,1000).eq.0) write(*,472) i
472    format('Still working...  Element ',i8)

c...fmlfm(i) is the flow (kg/s) between fracture and matrix
       fmlfm(i)=0.d0

c...jj is the number of connections for each element
       do jj=1,35
         flol(i,jj)=0.d0
c...icon(i,jj) is the node number of the element for connection jj to element i
         icon(i,jj)=0
       end do

      ELEMX=ELEMN(I)

c...If element is a boundary element, go to next element
      if(elemx(1:2).eq.'TP'.or.elemx(1:2).eq.'BT') go to 3000

c...Write the element number and the number of connections for that element
      if(i.gt.1) write(22,*) i-1,ncon(i-1)
c________________________________________________________________________
c...For each element, loop over all connections to determine if
c...the element is either the first or second element in each connection
c...nc is the number of connections per element

      nc=1
      DO 3001 J=1,NCMAX

c...Say element is the first element in the connection
       if(elem1(j).eq.elemx) then
        nsign=-1
c...If connecting element is the top boundary, go to next connection
        if(elem2(j)(1:2).eq.'TP') go to 3001
c...If connecting element is the bottom boundary, treat the flow to the
c...bottom boundary as a sink/source term and move on to the next connection
        if(elem2(j)(1:2).eq.'BT') then
          gelem(i)=fluxl(j)*nsign
          go to 3001
        end if
c...What is the second element in the connection?
        do ii=1,nmax
         if(elem2(j).eq.elemn(ii)) then
           k2nd=ii
c...Determine if the connection is between a fracture and matrix element
c...If it is a fracture-matrix connection (both elements have the same
c...coordinates, or ifm=2), store this flux separately from fracture-fracture
c...or matrix-matrix fluxes.
           dx=dabs(x(k2nd)-x(i))
           dy=dabs(y(k2nd)-y(i))
           dz=dabs(z(k2nd)-z(i))
           if(dx.le.zero.and.dy.le.zero.and.dz.le.zero.or.
     &        ifm(j).eq.2) then
c...If the first element of f-m connection is a fracture, then process this
             if(nfmc.eq.1) then
               go to 3017
             else
               go to 3001
             end if
           end if
           icon(i,nc)=ii
           flol(i,nc)=fluxl(j)*nsign
           nc=nc+1
           go to 3002
         endif
        end do
        write(*,7001) elemx,j,elem2(j),elem2(j-1),elem2(j+1)
7001    format('***Could not find 2nd element in connection for',
     &  ' first element ',a5,'***'/'Connection index = ',i8/
     &  'Second element = ',a5/'j-1= ',a5/'j+1= ',a5)
        stop
       end if

c...If no match in first element of connection, try second element
       if(elem2(j).eq.elemx) then
        nsign=1
c...If connecting element is the top boundary, go to next connection
        if(elem1(j)(1:2).eq.'TP') go to 3001
c...If connecting element is the bottom boundary, treat the flow to the
c...bottom boundary as a sink/source term and move on to the next connection
        if(elem1(j)(1:2).eq.'BT') then
          gelem(i)=fluxl(j)*nsign
          go to 3001
        end if
c...What is the first element in the connection?
        do ii=1,nmax
         if(elem1(j).eq.elemn(ii)) then
          k2nd=ii
c...Determine if the connection is between a fracture and matrix element
c...If it is a fracture-matrix connection (both elements have the same
c...coordinates), store this flux separately from fracture-fracture or
c...matrix-matrix fluxes.
           dx=dabs(x(k2nd)-x(i))
           dy=dabs(y(k2nd)-y(i))
           dz=dabs(z(k2nd)-z(i))
           if(dx.le.zero.and.dy.le.zero.and.dz.le.zero.or.
     &        ifm(j).eq.2) then
c...If the second element of f-m connection is a fracture, then process this
             if(nfmc.eq.2) then
               go to 3017
             else
               go to 3001
             end if
           end if
           icon(i,nc)=ii
           flol(i,nc)=fluxl(j)*nsign
           nc=nc+1
           go to 3002
         end if
        end do
        write(*,7000) elemx,j,elem1(j)
7000    format('***Could not find 1st element in connection for',
     &  ' second element ',a5,'***'/'Connection index = ',i8/
     &  '1st element = ',a5)
        stop
       end if

c...If neither element 1 or 2 for connection j is equal to elemx, then
c...go on to the next connection
       goto 3001

3002   continue

c_______________________________________________________________________
c...go to next connection
       go to 3001
c_______________________________________________________________________
c...Come here if this is a fracture-matrix connection AND the element
c...being considered (elemx=elemn(i)) is a fracture
c...Consider outflow to be positive and
c...that the first element in the connection is a fracture
3017  continue
      fmlfm(nmlfm)=nsign*fluxl(j)
      nmlfm=nmlfm+1

c...Go to next connection
c_______________________________________________________________________
3001  continue

c...ncon(i) is the total number of connections for node i
      ncon(i)=nc-1
C
c...Check
c        write(15,446) i,ncon(i),(icon(i,j),j=1,ncon(i))
c446     format(10(i8,2x))
c        write(15,448) i,ncon(i),(flol(i,j),j=1,ncon(i))
c448     format(2(i8,2x),8(e10.4,2x))
c...End check

c...Go to next element
3000  CONTINUE

c...nmlfm is the total number of fracture-matrix connections
```

```
      nmlfm=nmlfm-1

c...Add connection for each element to itself using generation array
c...nmfluxval is the total number of mass flux values
c...Note: nodes 1-nnodes are still assumed to alternative between
c...fractures and matrix. This will be adjusted later in the print-out
c...to the FEHM files.
      nmfluxval=0
      do i=1,nnodes
        ncon(i)=ncon(i)+1
        icon(i,ncon(i))=i
        flol(i,ncon(i))=gelem(i)
        nmfluxval=nmfluxval+ncon(i)
c...Check
c        write(15,448) i,ncon(i),flol(i,ncon(i)),nmfluxval
c448     format(2(i8,2x),e10.4,2x,i8)
c...End check
c...nmfluxval is the total number of flux values for fracture and matrix
c...elements excluding f-m fluxes
      end do

c...Call sort subroutine to sort the necessary arrays in ascending order
c...of elements for each connection pair of a given element

      call sort(nnodes)
C
c...Create 1-D arrays containing icon and flol information.  The arrays
c...will be icon2 and flol2. This assumes that the fractures and matrix
c...elements alternate in ELEME and fractures are listed first.
      k=1
      jj=1
      ncont1=0
c...ncont1 is the total number of connections for each continuum
c...do the fracture continuum first
      do i=1,nnodes,2
        do j=1,ncon(i)
c...The index k+nnodes/2+1 accounts for the leading pointer information
          icon2(k+nnodes/2+1)=(icon(i,j)+1)/2
          flol2(k)=flol(i,j)
          k=k+1
        end do
        ncont1=ncont1+ncon(i)
c...ncon2(jj) is the number of connections for fracture node jj, where jj is
c...now icremented 1,2,3...nnodes/2
        ncon2(jj)=ncon(i)
        jj=jj+1
      end do

c...Now do the matrix continuum
      do i=2,nnodes,2
        do j=1,ncon(i)
          flol2(k)=flol(i,j)
          k=k+1
        end do
      end do
c...ntotmfv is the total number of connections.  This can be compared to
c...nmfluxval as a cross-check to see if they're equal.
      ntotmfv=k-1

c...Write mass flux values to .ini file
      write(14,602) nmlfm+nmfluxval,ntotmfv,nnodes,nmlfm
602   format('mass flux values'/i8,5x,'#ntotmfv=',i8,', nnodes=',i8,
     & ', number of f-m connections= ',i8)
```

```
      write(14,604) (flol2(i),i=1,ntotmfv),(fmlfm(i),i=1,nmlfm)
604   format(5g15.8)

c...Write .stor file
      write(15,702) header
702   format(a80/'This is a .stor file with dummy area coefficients')

c...Add the pointer information (number of fracture nodes+1) to ncont1
      neq=nnodes/2
      ncont=ncont1+(neq+1)
      iwtotl=ncont-(neq+1)

      write(15,704) iwtotl,neq,ncont,1
704   format(4(i8,2x))

c...Write primary volume for each node to .stor
c...If this is an LBNL run, check to see how user wants calculation performed
c...If nprimary=1, simply sum the fracture and matrix volumes in ELEME to get
c...the primary volume.  If not, then divide the fracture volumes by the
c...fracture porosity, since the volumes in ELEME were multiplied by
c...the fracture porosity.
      if(nprimary.eq.1) then
        do i=1,nnodes,2
          vol(i)=vol(i)+vol(i+1)
        end do
        go to 835
      end if
      if(neos.eq.3.or.neos.eq.4) then
        do i=1,nnodes,2
          do j=1,ntotmat
            if(mat(i).eq.matname(j)) then
              vol(i)=vol(i)/por(j)
              go to 833
            end if
          end do
833     end do
      end if
c...If the fracture volumes were globally modified, multiply the volume
c...by a scaling factor, volscale, specified by the user to get the original
c...volume back.
835   write(15,706) (vol(i)*volscale,i=1,nnodes,2)
706   format(1p5e16.8)

c...Compile and write ncon and pointer information
c...Fill the icon2(i) array from i=1,neq+1 (recall that icon2(i) has
c...already been filled from neq+2 to ncont1 (the total number of connections
c...for the fracture continuum
      icon2(1)=neq+1
      do i=2,neq+1
        icon2(i)=icon2(i-1)+ncon2(i-1)
      end do
      write(15,708) (icon2(i),i=1,ncont)
708   format(5(i8,2x))

c...Compile and write istrw information to .stor file
      do i=1,ncont
        if(i.le.iwtotl) then
          istrw(i)=i
        else
          istrw(i)=0
        end if
      end do
      write(15,708) (istrw(i),i=1,ncont)
```

```
c...Compile and write nelmdg information to .stor file
      do i=1,neq
        do j=icon2(i)+1,icon2(i+1)
          if(icon2(j).eq.i) nelmdg(i)=j
        end do
      end do
      write(15,708) (nelmdg(i),i=1,neq)

c...Write dummy area coefficients to .stor file
      do i=1,3
        write(15,706) (-1.0,j=1,iwtotl)
      end do
c______________________________________________________________________

      write(*,1153) time
1153  format('Finished processing printout at ',e12.4,' sec')
      go to 722
C
90    CONTINUE
      write(*,*)'**Did not find desired print-out time in TOUGH2.OUT**'
C
722   write(*,*) 'Done!!!'

      stop
      END


      subroutine sort(nnodes)
c______________________________________________________________________
c  This subroutine sorts variables using a multipass method.
c     C.K.Ho
c     9/8/97
c______________________________________________________________________

      implicit double precision (a-h,o-z)
      common/int/ ncon(99000),icon(99000,35)
      common/flux/ flol(99000,35)

c...The objective here is to arrange the connections in ascending order
c...of connecting node number. The associated flux should also be sorted.

      nsort=1
      do i=1,nnodes
5       if(nsort.eq.1) then
          nsort=0
          do j=1,ncon(i)-1
            if(icon(i,j).gt.icon(i,j+1)) then
              itempicon=icon(i,j)
              icon(i,j)=icon(i,j+1)
              icon(i,j+1)=itempicon
              tempflol=flol(i,j)
              flol(i,j)=flol(i,j+1)
              flol(i,j+1)=tempflol
              nsort=1
            end if
          end do
          go to 5
        end if
        nsort=1
      end do
      return
      end
```

ATTACHMENT II

Listing of Software Routine WTRISE v. 1.0

```
program wtrise_v1
c_______________________________________________________________________
c  This program modifies the .ini restart file for FEHM to account for
c  a hypothetical water table rise in the LBNL site-scale flow fields.
c  The user inputs the prescribed water table elevation, and this
c  program will create user-prescribed (probably large ~1.e10) sink
c  values for those nodes that are located beneath the prescribed
c  water table elevation.  This will instantaneously remove
c  particles that are located "beneath" the water table.  The
c  liquid saturation of nodes beneath the water table is also changed
c  to a value of one.
c
c  (1) Read in user-defined files that contain the .grid, .stor, and
c      .ini files for FEHM, as well as the prescribed water-table rise
c      height (m) and "sink" value (kg/s)
c  (2) Read in nodes and coordinates from the .grid file
c  (3) Read in neq (number of fracture or matrix nodes), ncont, and
c      nelmdg values from .stor file.
c  (4) Read in .ini file and re-write new .ini file that has user-
c      prescribed sinks for nodes below prescribed water table height.
c      Also change saturations to one for nodes below water table.
c
c   C.K.Ho
c   10/13/1999
c_______________________________________________________________________
      character*80 grid,stor,ini,out,line
      character*90 line2
      allocatable :: x(:)
      allocatable :: y(:)
      allocatable :: z(:)
      allocatable :: vol(:)
      allocatable :: idum(:)
      allocatable :: nelmdg(:)
      allocatable :: sf(:)
      allocatable :: sm(:)
      allocatable :: pf(:)
      allocatable :: pm(:)
      allocatable :: ff(:)
      allocatable :: fm(:)
      allocatable :: ffm(:)

c  (1) Read in user-defined files that contain the .grid, .stor, and
c      .ini files for FEHM, as well as the prescribed water-table rise

      write(*,*)'What is the name of the .grid file?'
      read(*,*) grid
      write(*,*)'What is the name of the .stor file?'
      read(*,*) stor
      write(*,*)'What is the name of the .ini file?'
      read(*,*) ini
      write(*,*)'What is the desired water table elevation (m)?'
      read(*,*) wtr
      write(*,*)'What is the "sink" value for nodes below the'
      write(*,*)'water table (kg/s)?'
      read(*,*) sink
      write(*,*)'What is the desired name of the output file?'
      read(*,*) out
      write(*,*)'Print nodes below water table? (1=yes, 2=no)'
      read(*,*) ians

      open(10,file=grid,status='old')
      open(11,file=stor,status='old')
      open(12,file=ini,status='old')
      open(13,file=out,status='unknown')
      if(ians.eq.1) then
        write(*,*)'File will be called wtnodes.dat'
        open(14,file='wtnodes.dat',status='unknown')
      end if

c  (2) Read in nodes and coordinates from the .grid file

      read(10,*)
      read(10,*) nnodes
      allocate(x(nnodes),y(nnodes),z(nnodes))
      do i=1,nnodes
        read(10,*) j,x(i),y(i),z(i)
      end do

c  (3) Read in neq (number of fracture or matrix nodes), ncont, and
c      nelmdg values from .stor file.

      read(11,*)
      read(11,*)
      read(11,*) iwtotl,neq,ncont
      if(nnodes.ne.neq) write(*,*)'nnodes not equal to ncont!'

c  Allocate space for volumes and read in from .stor, but deallocate
c  space immediately afterwards since the volumes are not used.
c  They are read in just to get down to nelmdg.
      allocate(vol(neq))
      read(11,*) (vol(i),i=1,neq)
      deallocate(vol)

c  Allocate space for idum with size ncont and read in two arrays
c  with that size from .stor
      allocate(idum(ncont))
      read(11,*) (idum(i),i=1,ncont)
      read(11,*) (idum(i),i=1,ncont)

c  Now read in values for nelmdg(i), which is the position (index)
c  of node i in the ncon array
      allocate(nelmdg(neq))
      read(11,*) (nelmdg(i),i=1,neq)

c  (4) Read in .ini file and re-write new .ini file that has user-
c      prescribed sinks for nodes below prescribed water table height.
c      Also change saturations to one for nodes below water table.

c  Read in header information
      read(12,'(a)') line
      write(13,'(a)') line
      read(12,'(a)') line
      write(13,20) wtr,line
20    format('WTR= ',f5.1,' m. ',a80)
      do i=1,6
        read(12,'(a)') line
        write(13,'(a)') line
      end do

c  Read in saturations (fractures first then matrix).  If node is
```

```
c  below water table, change saturation to one.
      allocate(sf(neq))
      allocate(sm(neq))
      read(12,40) (sf(i),i=1,neq),(sm(i),i=1,neq)
      icheck=0
      do i=1,neq
        if(z(i).le.wtr) then
          sf(i)=1.
          sm(i)=1.
          icheck=icheck+1
        end if
      end do
      write(13,40) (sf(i),i=1,neq),(sm(i),i=1,neq)
40    format(4g16.8)

c  Read in pressures (fractures first then matrix).
      allocate(pf(neq))
      allocate(pm(neq))
      read(12,40) (pf(i),i=1,neq),(pm(i),i=1,neq)
      write(13,40) (pf(i),i=1,neq),(pm(i),i=1,neq)

c  Read in mass flux values (fractures first then matrix)
      allocate(ff(iwtotl))
      allocate(fm(iwtotl))
      allocate(ffm(neq))
      read(12,'(a)') line
      write(13,'(a)') line
      read(12,'(a)') line2
      write(13,'(a)') line2

c  Note that iwtotl=ncont-neq-1
      read(12,50) (ff(i),i=1,iwtotl),(fm(i),i=1,iwtotl),
     &            (ffm(i),i=1,neq)
50    format(5g15.8)

c  Go through each connection and modify sources for those
c  nodes that are below the water table. j is the node number.
c  i denotes the index in the ncon array starting with 1 instead
c  of neq+1.
      j=1
      icheckf=0
      do i=1,iwtotl
        if(i.eq.nelmdg(j)-neq-1) then
          if(z(j).le.wtr) then
            ff(i)=sink
            fm(i)=sink
            icheckf=icheckf+1
          end if
          j=j+1
        end if
      end do

      write(*,55) icheck
55    format('Number of saturations per continuum modified = ',i7)
      write(*,60) icheckf
60    format('Number of source nodes per continuum modified = ',i7)

      write(13,50) (ff(i),i=1,iwtotl),(fm(i),i=1,iwtotl),
     &            (ffm(i),i=1,neq)

c  Write "check" information to file if user desired
      if(ians.eq.1) then
      write(14,70)
```

```
70    format('node, x(m), y(m), z(m), sf, sm, ff(kg/s), fm(kg/s)')
        do i=1,neq
          if(z(i).le.wtr) then
            write(14,80)i,x(i),y(i),z(i),sf(i),sm(i),
     &      ff(nelmdg(i)-neq-1),fm(nelmdg(i)-neq-1)
80          format(i6,3(', ',f10.1),4(', ',e11.4))
          end if
        end do
      end if

      stop
      end
```

ATTACHMENT III

Directory of Files Submitted to Technical Data Management System (DTN: SN9910T0581699.002)

```
a U0125_fehm_files/ 0K
a U0125_fehm_files/fm_glal1/ 0K
a U0125_fehm_files/fm_glal1/README 2K
a U0125_fehm_files/fm_glal1/SR-repo-nodes 2K
a U0125_fehm_files/fm_glal1/fm_glal1.check 1114K
a U0125_fehm_files/fm_glal1/fm_glal1.dat 1K
a U0125_fehm_files/fm_glal1/fm_glal1.dpdp 2K
a U0125_fehm_files/fm_glal1/fm_glal1.files 1K
a U0125_fehm_files/fm_glal1/fm_glal1.grid 2235K
a U0125_fehm_files/fm_glal1/fm_glal1.ini 15356K
a U0125_fehm_files/fm_glal1/fm_glal1.rock 7K
a U0125_fehm_files/fm_glal1/fm_glal1.stor 28196K
a U0125_fehm_files/fm_glal1/fm_glal1.zone 946K
a U0125_fehm_files/fm_glal1/fm_glal1.zone2 952K
a U0125_fehm_files/fm_glal1/fm_glal1_wt850.ini 15356K
a U0125_fehm_files/fm_glal1/README_wtrise 1K
a U0125_fehm_files/fm_glal1/runt2fehm2 1K
a U0125_fehm_files/fm_glal1/runwtrise 1K
a U0125_fehm_files/fm_glal1/t2fehm2.inp 1K
a U0125_fehm_files/fm_glal1/t2fehm2.out 5K
a U0125_fehm_files/fm_glal1/wtrise.inp 1K
a U0125_fehm_files/fm_glal1/wtrise.out 1K
a U0125_fehm_files/fm_glal2/ 0K
a U0125_fehm_files/fm_glal2/README 2K
a U0125_fehm_files/fm_glal2/SR-repo-nodes 2K
a U0125_fehm_files/fm_glal2/fm_glal2.check 1114K
a U0125_fehm_files/fm_glal2/fm_glal2.dat 1K
a U0125_fehm_files/fm_glal2/fm_glal2.dpdp 2K
a U0125_fehm_files/fm_glal2/fm_glal2.files 1K
a U0125_fehm_files/fm_glal2/fm_glal2.grid 2235K
a U0125_fehm_files/fm_glal2/fm_glal2.ini 15356K
a U0125_fehm_files/fm_glal2/fm_glal2.rock 7K
a U0125_fehm_files/fm_glal2/fm_glal2.stor 28196K
a U0125_fehm_files/fm_glal2/fm_glal2.zone 946K
a U0125_fehm_files/fm_glal2/fm_glal2.zone2 952K
a U0125_fehm_files/fm_glal2/fm_glal2_wt850.ini 15356K
a U0125_fehm_files/fm_glal2/README_wtrise 1K
a U0125_fehm_files/fm_glal2/runt2fehm2 1K
a U0125_fehm_files/fm_glal2/runwtrise 1K
a U0125_fehm_files/fm_glal2/t2fehm2.inp 1K
a U0125_fehm_files/fm_glal2/t2fehm2.out 5K
a U0125_fehm_files/fm_glal2/wtrise.inp 1K
a U0125_fehm_files/fm_glal2/wtrise.out 1K
a U0125_fehm_files/fm_glam1/ 0K
a U0125_fehm_files/fm_glam1/README 2K
a U0125_fehm_files/fm_glam1/SR-repo-nodes 2K
a U0125_fehm_files/fm_glam1/glam1_post/ 0K
a U0125_fehm_files/fm_glam1/glam1_post/glam1.ini2 15356K
a U0125_fehm_files/fm_glam1/glam1_post/glam1.rock2 7K
a U0125_fehm_files/fm_glam1/glam1_post/glam1.stor2 28196K
a U0125_fehm_files/fm_glam1/glam1_post/glam1_inf.dat 78K
a U0125_fehm_files/fm_glam1/glam1_post/glam1_rep_perc.dat.qf 1154K
a U0125_fehm_files/fm_glam1/glam1_post/glam1_rep_perc.dat.qm 1154K
a U0125_fehm_files/fm_glam1/glam1_post/glam1_rep_perc.dat.tot 848K
a U0125_fehm_files/fm_glam1/glam1_post/glam1_rep_perc.dat.x 75K
a U0125_fehm_files/fm_glam1/glam1_post/glam1_rep_perc2.dat.qf 2584K
```

```
a U0125_fehm_files/fm_glam1/glam1_post/glam1_rep_perc2.dat.qm 2584K
a U0125_fehm_files/fm_glam1/glam1_post/glam1_rep_perc2.dat.to 1899K
a U0125_fehm_files/fm_glam1/glam1_post/glam1_wt.dat 285K
a U0125_fehm_files/fm_glam1/glam1_post/glam1_wtperc.dat 135K
a U0125_fehm_files/fm_glam1/glam1_post/t2postgen.inp 1K
a U0125_fehm_files/fm_glam1/glam1_post/t2postgen_v2 28K
a U0125_fehm_files/fm_glam1/glam1_post/t2postgen_v2.f 5K
a U0125_fehm_files/fm_glam1/glam1_post/t2postrep.inp 1K
a U0125_fehm_files/fm_glam1/glam1_post/t2postrep.out 1K
a U0125_fehm_files/fm_glam1/glam1_post/t2postrep_v2 33K
a U0125_fehm_files/fm_glam1/glam1_post/t2postrep_v2.f 9K
a U0125_fehm_files/fm_glam1/glam1_post/t2postwt.inp 1K
a U0125_fehm_files/fm_glam1/glam1_post/t2postwt_v2 22K
a U0125_fehm_files/fm_glam1/glam1_post/t2postwt_v2.f 4K
a U0125_fehm_files/fm_glam1/glam1_post/grep_wt 1K
a U0125_fehm_files/fm_glam1/glam1_post/extractz 18K
a U0125_fehm_files/fm_glam1/glam1_post/extractz.f 2K
a U0125_fehm_files/fm_glam1/glam1_post/README.2 2K
a U0125_fehm_files/fm_glam1/fm_glam1.check 1114K
a U0125_fehm_files/fm_glam1/fm_glam1.dat 1K
a U0125_fehm_files/fm_glam1/fm_glam1.dpdp 2K
a U0125_fehm_files/fm_glam1/fm_glam1.files 1K
a U0125_fehm_files/fm_glam1/fm_glam1.grid 2235K
a U0125_fehm_files/fm_glam1/fm_glam1.ini 15356K
a U0125_fehm_files/fm_glam1/fm_glam1.rock 7K
a U0125_fehm_files/fm_glam1/fm_glam1.stor 28196K
a U0125_fehm_files/fm_glam1/fm_glam1.zone 946K
a U0125_fehm_files/fm_glam1/fm_glam1.zone2 952K
a U0125_fehm_files/fm_glam1/fm_glam1_wt850.ini 15356K
a U0125_fehm_files/fm_glam1/README_wtrise 1K
a U0125_fehm_files/fm_glam1/runt2fehm2 1K
a U0125_fehm_files/fm_glam1/runwtrise 1K
a U0125_fehm_files/fm_glam1/t2fehm2.inp 1K
a U0125_fehm_files/fm_glam1/t2fehm2.out 5K
a U0125_fehm_files/fm_glam1/wtrise.inp 1K
a U0125_fehm_files/fm_glam1/wtrise.out 1K
a U0125_fehm_files/fm_glam2/ 0K
a U0125_fehm_files/fm_glam2/README 2K
a U0125_fehm_files/fm_glam2/SR-repo-nodes 2K
a U0125_fehm_files/fm_glam2/fm_glam2.check 1114K
a U0125_fehm_files/fm_glam2/fm_glam2.dat 1K
a U0125_fehm_files/fm_glam2/fm_glam2.dpdp 2K
a U0125_fehm_files/fm_glam2/fm_glam2.files 1K
a U0125_fehm_files/fm_glam2/fm_glam2.grid 2235K
a U0125_fehm_files/fm_glam2/fm_glam2.ini 15356K
a U0125_fehm_files/fm_glam2/fm_glam2.rock 7K
a U0125_fehm_files/fm_glam2/fm_glam2.stor 28196K
a U0125_fehm_files/fm_glam2/fm_glam2.zone 946K
a U0125_fehm_files/fm_glam2/fm_glam2.zone2 952K
a U0125_fehm_files/fm_glam2/fm_glam2_wt850.ini 15356K
a U0125_fehm_files/fm_glam2/README_wtrise 1K
a U0125_fehm_files/fm_glam2/runt2fehm2 1K
a U0125_fehm_files/fm_glam2/runwtrise 1K
a U0125_fehm_files/fm_glam2/t2fehm2.inp 1K
a U0125_fehm_files/fm_glam2/t2fehm2.out 5K
a U0125_fehm_files/fm_glam2/t2fehm2_v2 74K
a U0125_fehm_files/fm_glam2/wtrise.inp 1K
a U0125_fehm_files/fm_glam2/wtrise.out 1K
a U0125_fehm_files/fm_glau1/ 0K
a U0125_fehm_files/fm_glau1/README 2K
a U0125_fehm_files/fm_glau1/SR-repo-nodes 2K
a U0125_fehm_files/fm_glau1/fm_glau1.check 1114K
a U0125_fehm_files/fm_glau1/fm_glau1.dat 1K
```

```
a U0125_fehm_files/fm_glau1/fm_glau1.dpdp 2K
a U0125_fehm_files/fm_glau1/fm_glau1.files 1K
a U0125_fehm_files/fm_glau1/fm_glau1.grid 2235K
a U0125_fehm_files/fm_glau1/fm_glau1.ini 15356K
a U0125_fehm_files/fm_glau1/fm_glau1.rock 7K
a U0125_fehm_files/fm_glau1/fm_glau1.stor 28196K
a U0125_fehm_files/fm_glau1/fm_glau1.zone 946K
a U0125_fehm_files/fm_glau1/fm_glau1.zone2 952K
a U0125_fehm_files/fm_glau1/fm_glau1_wt850.ini 15356K
a U0125_fehm_files/fm_glau1/README_wtrise 1K
a U0125_fehm_files/fm_glau1/runt2fehm2 1K
a U0125_fehm_files/fm_glau1/runwtrise 1K
a U0125_fehm_files/fm_glau1/t2fehm2.inp 1K
a U0125_fehm_files/fm_glau1/t2fehm2.out 5K
a U0125_fehm_files/fm_glau1/wtrise.inp 1K
a U0125_fehm_files/fm_glau1/wtrise.out 1K
a U0125_fehm_files/fm_glau2/ 0K
a U0125_fehm_files/fm_glau2/README 2K
a U0125_fehm_files/fm_glau2/SR-repo-nodes 2K
a U0125_fehm_files/fm_glau2/fm_glau2.check 1114K
a U0125_fehm_files/fm_glau2/fm_glau2.dat 1K
a U0125_fehm_files/fm_glau2/fm_glau2.dpdp 2K
a U0125_fehm_files/fm_glau2/fm_glau2.files 1K
a U0125_fehm_files/fm_glau2/fm_glau2.grid 2235K
a U0125_fehm_files/fm_glau2/fm_glau2.ini 15356K
a U0125_fehm_files/fm_glau2/fm_glau2.rock 7K
a U0125_fehm_files/fm_glau2/fm_glau2.stor 28196K
a U0125_fehm_files/fm_glau2/fm_glau2.zone 946K
a U0125_fehm_files/fm_glau2/fm_glau2.zone2 952K
a U0125_fehm_files/fm_glau2/fm_glau2_wt850.ini 15356K
a U0125_fehm_files/fm_glau2/README_wtrise 1K
a U0125_fehm_files/fm_glau2/runt2fehm2 1K
a U0125_fehm_files/fm_glau2/runwtrise 1K
a U0125_fehm_files/fm_glau2/t2fehm2.inp 1K
a U0125_fehm_files/fm_glau2/t2fehm2.out 5K
a U0125_fehm_files/fm_glau2/wtrise.inp 1K
a U0125_fehm_files/fm_glau2/wtrise.out 1K
a U0125_fehm_files/fm_monl1/ 0K
a U0125_fehm_files/fm_monl1/README 2K
a U0125_fehm_files/fm_monl1/SR-repo-nodes 2K
a U0125_fehm_files/fm_monl1/fm_monl1.check 1114K
a U0125_fehm_files/fm_monl1/fm_monl1.dat 1K
a U0125_fehm_files/fm_monl1/fm_monl1.dpdp 2K
a U0125_fehm_files/fm_monl1/fm_monl1.files 1K
a U0125_fehm_files/fm_monl1/fm_monl1.grid 2235K
a U0125_fehm_files/fm_monl1/fm_monl1.ini 15356K
a U0125_fehm_files/fm_monl1/fm_monl1.rock 7K
a U0125_fehm_files/fm_monl1/fm_monl1.stor 28196K
a U0125_fehm_files/fm_monl1/fm_monl1.zone 946K
a U0125_fehm_files/fm_monl1/fm_monl1.zone2 952K
a U0125_fehm_files/fm_monl1/fm_monl1_wt850.ini 15356K
a U0125_fehm_files/fm_monl1/README_wtrise 1K
a U0125_fehm_files/fm_monl1/runt2fehm2 1K
a U0125_fehm_files/fm_monl1/runwtrise 1K
a U0125_fehm_files/fm_monl1/t2fehm2.inp 1K
a U0125_fehm_files/fm_monl1/t2fehm2.out 5K
a U0125_fehm_files/fm_monl1/wtrise.inp 1K
a U0125_fehm_files/fm_monl1/wtrise.out 1K
a U0125_fehm_files/fm_monl2/ 0K
a U0125_fehm_files/fm_monl2/README 2K
a U0125_fehm_files/fm_monl2/SR-repo-nodes 2K
a U0125_fehm_files/fm_monl2/fm_monl2.check 1114K
a U0125_fehm_files/fm_monl2/fm_monl2.dat 1K
```

```
a U0125_fehm_files/fm_monl2/fm_monl2.dpdp 2K
a U0125_fehm_files/fm_monl2/fm_monl2.files 1K
a U0125_fehm_files/fm_monl2/fm_monl2.grid 2235K
a U0125_fehm_files/fm_monl2/fm_monl2.ini 15356K
a U0125_fehm_files/fm_monl2/fm_monl2.rock 7K
a U0125_fehm_files/fm_monl2/fm_monl2.stor 28196K
a U0125_fehm_files/fm_monl2/fm_monl2.zone 946K
a U0125_fehm_files/fm_monl2/fm_monl2.zone2 952K
a U0125_fehm_files/fm_monl2/fm_monl2_wt850.ini 15356K
a U0125_fehm_files/fm_monl2/README_wtrise 1K
a U0125_fehm_files/fm_monl2/runt2fehm2 1K
a U0125_fehm_files/fm_monl2/runwtrise 1K
a U0125_fehm_files/fm_monl2/t2fehm2.inp 1K
a U0125_fehm_files/fm_monl2/t2fehm2.out 5K
a U0125_fehm_files/fm_monl2/wtrise.inp 1K
a U0125_fehm_files/fm_monl2/wtrise.out 1K
a U0125_fehm_files/fm_monm1/ 0K
a U0125_fehm_files/fm_monm1/README 2K
a U0125_fehm_files/fm_monm1/SR-repo-nodes 2K
a U0125_fehm_files/fm_monm1/fm_monm1.check 1114K
a U0125_fehm_files/fm_monm1/fm_monm1.dat 1K
a U0125_fehm_files/fm_monm1/fm_monm1.dpdp 2K
a U0125_fehm_files/fm_monm1/fm_monm1.files 1K
a U0125_fehm_files/fm_monm1/fm_monm1.grid 2235K
a U0125_fehm_files/fm_monm1/fm_monm1.ini 15356K
a U0125_fehm_files/fm_monm1/fm_monm1.rock 7K
a U0125_fehm_files/fm_monm1/fm_monm1.stor 28196K
a U0125_fehm_files/fm_monm1/fm_monm1.zone 946K
a U0125_fehm_files/fm_monm1/fm_monm1.zone2 952K
a U0125_fehm_files/fm_monm1/fm_monm1_wt850.ini 15356K
a U0125_fehm_files/fm_monm1/README_wtrise 1K
a U0125_fehm_files/fm_monm1/runt2fehm2 1K
a U0125_fehm_files/fm_monm1/runwtrise 1K
a U0125_fehm_files/fm_monm1/t2fehm2.inp 1K
a U0125_fehm_files/fm_monm1/t2fehm2.out 5K
a U0125_fehm_files/fm_monm1/wtrise.inp 1K
a U0125_fehm_files/fm_monm1/wtrise.out 1K
a U0125_fehm_files/fm_monm2/ 0K
a U0125_fehm_files/fm_monm2/README 2K
a U0125_fehm_files/fm_monm2/SR-repo-nodes 2K
a U0125_fehm_files/fm_monm2/fm_monm2.check 1114K
a U0125_fehm_files/fm_monm2/fm_monm2.dat 1K
a U0125_fehm_files/fm_monm2/fm_monm2.dpdp 2K
a U0125_fehm_files/fm_monm2/fm_monm2.files 1K
a U0125_fehm_files/fm_monm2/fm_monm2.grid 2235K
a U0125_fehm_files/fm_monm2/fm_monm2.ini 15356K
a U0125_fehm_files/fm_monm2/fm_monm2.rock 7K
a U0125_fehm_files/fm_monm2/fm_monm2.stor 28196K
a U0125_fehm_files/fm_monm2/fm_monm2.zone 946K
a U0125_fehm_files/fm_monm2/fm_monm2.zone2 952K
a U0125_fehm_files/fm_monm2/fm_monm2_wt850.ini 15356K
a U0125_fehm_files/fm_monm2/README_wtrise 1K
a U0125_fehm_files/fm_monm2/runt2fehm2 1K
a U0125_fehm_files/fm_monm2/runwtrise 1K
a U0125_fehm_files/fm_monm2/t2fehm2.inp 1K
a U0125_fehm_files/fm_monm2/t2fehm2.out 5K
a U0125_fehm_files/fm_monm2/wtrise.inp 1K
a U0125_fehm_files/fm_monm2/wtrise.out 1K
a U0125_fehm_files/fm_monu1/ 0K
a U0125_fehm_files/fm_monu1/README 2K
a U0125_fehm_files/fm_monu1/SR-repo-nodes 2K
a U0125_fehm_files/fm_monu1/fm_monu1.check 1114K
a U0125_fehm_files/fm_monu1/fm_monu1.dat 1K
```

```
a U0125_fehm_files/fm_monu1/fm_monu1.dpdp 2K
a U0125_fehm_files/fm_monu1/fm_monu1.files 1K
a U0125_fehm_files/fm_monu1/fm_monu1.grid 2235K
a U0125_fehm_files/fm_monu1/fm_monu1.ini 15356K
a U0125_fehm_files/fm_monu1/fm_monu1.rock 7K
a U0125_fehm_files/fm_monu1/fm_monu1.stor 28196K
a U0125_fehm_files/fm_monu1/fm_monu1.zone 946K
a U0125_fehm_files/fm_monu1/fm_monu1.zone2 952K
a U0125_fehm_files/fm_monu1/fm_monu1_wt850.ini 15356K
a U0125_fehm_files/fm_monu1/README_wtrise 1K
a U0125_fehm_files/fm_monu1/runt2fehm2 1K
a U0125_fehm_files/fm_monu1/runwtrise 1K
a U0125_fehm_files/fm_monu1/t2fehm2.inp 1K
a U0125_fehm_files/fm_monu1/t2fehm2.out 5K
a U0125_fehm_files/fm_monu1/wtrise.inp 1K
a U0125_fehm_files/fm_monu1/wtrise.out 1K
a U0125_fehm_files/fm_monu2/ 0K
a U0125_fehm_files/fm_monu2/README 2K
a U0125_fehm_files/fm_monu2/SR-repo-nodes 2K
a U0125_fehm_files/fm_monu2/fm_monu2.check 1114K
a U0125_fehm_files/fm_monu2/fm_monu2.dat 1K
a U0125_fehm_files/fm_monu2/fm_monu2.dpdp 2K
a U0125_fehm_files/fm_monu2/fm_monu2.files 1K
a U0125_fehm_files/fm_monu2/fm_monu2.grid 2235K
a U0125_fehm_files/fm_monu2/fm_monu2.ini 15356K
a U0125_fehm_files/fm_monu2/fm_monu2.rock 7K
a U0125_fehm_files/fm_monu2/fm_monu2.stor 28196K
a U0125_fehm_files/fm_monu2/fm_monu2.zone 946K
a U0125_fehm_files/fm_monu2/fm_monu2.zone2 952K
a U0125_fehm_files/fm_monu2/fm_monu2_wt850.ini 15356K
a U0125_fehm_files/fm_monu2/README_wtrise 1K
a U0125_fehm_files/fm_monu2/runt2fehm2 1K
a U0125_fehm_files/fm_monu2/runwtrise 1K
a U0125_fehm_files/fm_monu2/t2fehm2.inp 1K
a U0125_fehm_files/fm_monu2/t2fehm2.out 5K
a U0125_fehm_files/fm_monu2/wtrise.inp 1K
a U0125_fehm_files/fm_monu2/wtrise.out 1K
a U0125_fehm_files/fm_pchl1/ 0K
a U0125_fehm_files/fm_pchl1/fm_pchl1.check 1114K
a U0125_fehm_files/fm_pchl1/fm_pchl1.dat 1K
a U0125_fehm_files/fm_pchl1/fm_pchl1.dpdp 2K
a U0125_fehm_files/fm_pchl1/fm_pchl1.files 1K
a U0125_fehm_files/fm_pchl1/fm_pchl1.grid 2235K
a U0125_fehm_files/fm_pchl1/fm_pchl1.ini 15356K
a U0125_fehm_files/fm_pchl1/fm_pchl1.rock 7K
a U0125_fehm_files/fm_pchl1/fm_pchl1.stor 28196K
a U0125_fehm_files/fm_pchl1/fm_pchl1.zone 946K
a U0125_fehm_files/fm_pchl1/fm_pchl1.zone2 952K
a U0125_fehm_files/fm_pchl1/fm_pchl1_wt850.ini 15356K
a U0125_fehm_files/fm_pchl1/runt2fehm2 1K
a U0125_fehm_files/fm_pchl1/runwtrise 1K
a U0125_fehm_files/fm_pchl1/t2fehm2.inp 1K
a U0125_fehm_files/fm_pchl1/t2fehm2.out 5K
a U0125_fehm_files/fm_pchl1/wtrise.inp 1K
a U0125_fehm_files/fm_pchl1/wtrise.out 1K
a U0125_fehm_files/fm_pchl1/wtnodes.dat 888K
a U0125_fehm_files/fm_pchl1/README 2K
a U0125_fehm_files/fm_pchl1/SR-repo-nodes 2K
a U0125_fehm_files/fm_pchl1/README_wtrise 1K
a U0125_fehm_files/fm_pchl2/ 0K
a U0125_fehm_files/fm_pchl2/fm_pchl2.check 1114K
a U0125_fehm_files/fm_pchl2/fm_pchl2.dat 1K
a U0125_fehm_files/fm_pchl2/fm_pchl2.dpdp 2K
```

```
a U0125_fehm_files/fm_pchl2/fm_pchl2.files 1K
a U0125_fehm_files/fm_pchl2/fm_pchl2.grid 2235K
a U0125_fehm_files/fm_pchl2/fm_pchl2.ini 15356K
a U0125_fehm_files/fm_pchl2/fm_pchl2.rock 7K
a U0125_fehm_files/fm_pchl2/fm_pchl2.stor 28196K
a U0125_fehm_files/fm_pchl2/fm_pchl2.zone 946K
a U0125_fehm_files/fm_pchl2/fm_pchl2.zone2 952K
a U0125_fehm_files/fm_pchl2/fm_pchl2_wt850.ini 15356K
a U0125_fehm_files/fm_pchl2/README 2K
a U0125_fehm_files/fm_pchl2/SR-repo-nodes 2K
a U0125_fehm_files/fm_pchl2/README_wtrise 1K
a U0125_fehm_files/fm_pchl2/runt2fehm2 1K
a U0125_fehm_files/fm_pchl2/runwtrise 1K
a U0125_fehm_files/fm_pchl2/t2fehm2.inp 1K
a U0125_fehm_files/fm_pchl2/t2fehm2.out 5K
a U0125_fehm_files/fm_pchl2/wtrise.inp 1K
a U0125_fehm_files/fm_pchl2/wtrise.out 1K
a U0125_fehm_files/fm_pchm1/ 0K
a U0125_fehm_files/fm_pchm1/README 2K
a U0125_fehm_files/fm_pchm1/SR-repo-nodes 2K
a U0125_fehm_files/fm_pchm1/fm_pchm1.check 1114K
a U0125_fehm_files/fm_pchm1/fm_pchm1.dat 1K
a U0125_fehm_files/fm_pchm1/fm_pchm1.dpdp 2K
a U0125_fehm_files/fm_pchm1/fm_pchm1.files 1K
a U0125_fehm_files/fm_pchm1/fm_pchm1.grid 2235K
a U0125_fehm_files/fm_pchm1/fm_pchm1.ini 15356K
a U0125_fehm_files/fm_pchm1/fm_pchm1.rock 7K
a U0125_fehm_files/fm_pchm1/fm_pchm1.stor 28196K
a U0125_fehm_files/fm_pchm1/fm_pchm1.zone 946K
a U0125_fehm_files/fm_pchm1/fm_pchm1.zone2 952K
a U0125_fehm_files/fm_pchm1/fm_pchm1_wt850.ini 15356K
a U0125_fehm_files/fm_pchm1/README_wtrise 1K
a U0125_fehm_files/fm_pchm1/runt2fehm2 1K
a U0125_fehm_files/fm_pchm1/runwtrise 1K
a U0125_fehm_files/fm_pchm1/t2fehm2.inp 1K
a U0125_fehm_files/fm_pchm1/t2fehm2.out 5K
a U0125_fehm_files/fm_pchm1/wtrise.inp 1K
a U0125_fehm_files/fm_pchm1/wtrise.out 1K
a U0125_fehm_files/fm_pchm2/ 0K
a U0125_fehm_files/fm_pchm2/README 2K
a U0125_fehm_files/fm_pchm2/SR-repo-nodes 2K
a U0125_fehm_files/fm_pchm2/fm_pchm2.check 1114K
a U0125_fehm_files/fm_pchm2/fm_pchm2.dat 1K
a U0125_fehm_files/fm_pchm2/fm_pchm2.dpdp 2K
a U0125_fehm_files/fm_pchm2/fm_pchm2.files 1K
a U0125_fehm_files/fm_pchm2/fm_pchm2.grid 2235K
a U0125_fehm_files/fm_pchm2/fm_pchm2.ini 15356K
a U0125_fehm_files/fm_pchm2/fm_pchm2.rock 7K
a U0125_fehm_files/fm_pchm2/fm_pchm2.stor 28196K
a U0125_fehm_files/fm_pchm2/fm_pchm2.zone 946K
a U0125_fehm_files/fm_pchm2/fm_pchm2.zone2 952K
a U0125_fehm_files/fm_pchm2/fm_pchm2_wt850.ini 15356K
a U0125_fehm_files/fm_pchm2/README_wtrise 1K
a U0125_fehm_files/fm_pchm2/runt2fehm2 1K
a U0125_fehm_files/fm_pchm2/runwtrise 1K
a U0125_fehm_files/fm_pchm2/t2fehm2.inp 1K
a U0125_fehm_files/fm_pchm2/t2fehm2.out 5K
a U0125_fehm_files/fm_pchm2/t2fehm2_v2 74K
a U0125_fehm_files/fm_pchm2/wtrise.inp 1K
a U0125_fehm_files/fm_pchm2/wtrise.out 1K
a U0125_fehm_files/fm_pchu1/ 0K
a U0125_fehm_files/fm_pchu1/README 2K
a U0125_fehm_files/fm_pchu1/SR-repo-nodes 2K
```

```
a U0125_fehm_files/fm_pchu1/fm_pchu1.check 1114K
a U0125_fehm_files/fm_pchu1/fm_pchu1.dat 1K
a U0125_fehm_files/fm_pchu1/fm_pchu1.dpdp 2K
a U0125_fehm_files/fm_pchu1/fm_pchu1.files 1K
a U0125_fehm_files/fm_pchu1/fm_pchu1.grid 2235K
a U0125_fehm_files/fm_pchu1/fm_pchu1.ini 15356K
a U0125_fehm_files/fm_pchu1/fm_pchu1.rock 7K
a U0125_fehm_files/fm_pchu1/fm_pchu1.stor 28196K
a U0125_fehm_files/fm_pchu1/fm_pchu1.zone 946K
a U0125_fehm_files/fm_pchu1/fm_pchu1.zone2 952K
a U0125_fehm_files/fm_pchu1/fm_pchu1_wt850.ini 15356K
a U0125_fehm_files/fm_pchu1/README_wtrise 1K
a U0125_fehm_files/fm_pchu1/runt2fehm2 1K
a U0125_fehm_files/fm_pchu1/runwtrise 1K
a U0125_fehm_files/fm_pchu1/t2fehm2.inp 1K
a U0125_fehm_files/fm_pchu1/t2fehm2.out 5K
a U0125_fehm_files/fm_pchu1/wtrise.inp 1K
a U0125_fehm_files/fm_pchu1/wtrise.out 1K
a U0125_fehm_files/fm_pchu2/ 0K
a U0125_fehm_files/fm_pchu2/README 2K
a U0125_fehm_files/fm_pchu2/SR-repo-nodes 2K
a U0125_fehm_files/fm_pchu2/fm_pchu2.check 1114K
a U0125_fehm_files/fm_pchu2/fm_pchu2.dat 1K
a U0125_fehm_files/fm_pchu2/fm_pchu2.dpdp 2K
a U0125_fehm_files/fm_pchu2/fm_pchu2.files 1K
a U0125_fehm_files/fm_pchu2/fm_pchu2.grid 2235K
a U0125_fehm_files/fm_pchu2/fm_pchu2.ini 15356K
a U0125_fehm_files/fm_pchu2/fm_pchu2.rock 7K
a U0125_fehm_files/fm_pchu2/fm_pchu2.stor 28196K
a U0125_fehm_files/fm_pchu2/fm_pchu2.zone 946K
a U0125_fehm_files/fm_pchu2/fm_pchu2.zone2 952K
a U0125_fehm_files/fm_pchu2/fm_pchu2_wt850.ini 15356K
a U0125_fehm_files/fm_pchu2/README_wtrise 1K
a U0125_fehm_files/fm_pchu2/runt2fehm2 1K
a U0125_fehm_files/fm_pchu2/runwtrise 1K
a U0125_fehm_files/fm_pchu2/t2fehm2.inp 1K
a U0125_fehm_files/fm_pchu2/t2fehm2.out 5K
a U0125_fehm_files/fm_pchu2/wtrise.inp 1K
a U0125_fehm_files/fm_pchu2/wtrise.out 1K
a U0125_fehm_files/src/ 0K
a U0125_fehm_files/src/README 1K
a U0125_fehm_files/src/t2fehm2_v3 74K
a U0125_fehm_files/src/t2fehm2_v3.f 34K
a U0125_fehm_files/src/wtrise_v1 357K
a U0125_fehm_files/src/wtrise_v1.f 7K
a U0125_fehm_files/README 7K
a U0125_fehm_files/rock_files/ 0K
a U0125_fehm_files/rock_files/pch1.zone 4K
a U0125_fehm_files/rock_files/pch2.zone 3K
a U0125_fehm_files/rock_files/pch2.rock 3K
a U0125_fehm_files/rock_files/README 2K
a U0125_fehm_files/rock_files/pch1.rock 4K
```